MITSUBISHI ELECTRIC
Changes for the Better

Serie System Q - Manual Práctico

Versión 1.0

• PRECAUCIONES DE SEGURIDAD •

(Leer siempre estas advertencias antes de utilizar el producto)

Antes de utilizar este producto, por favor lea este manual y los manuales referentes en este manual,
Y mostrar total atención a los consejos para realizar un buen uso del producto.
Por favor mantenga este manual en un lugar seguro y accesible para cuando sea necesario.
Entregar siempre una copia de este manual al usuario final.

A - 1

REVISIONES

* El número de manual esta en el fondo izquierdo de la contraportada.

Fecha de Impresión	*Número de Manual	Revisión
Dic., 1999	SH (NA)-080039-A	Primera Edición
Jun., 2000	SH (NA)-080039-B	*Adición* APPENDIX5 *Corrección* CONTIENE, Sección 3.4, 6.6.1, 6.8.6, 6.8.8, 6.8.9, 7.6.8, 9.8, 10.3, 11.2, APP 1.2, APP 4
Sep., 2000	SH (NA)-080039-C	*Adición* Sección 9.9, 9.10, 9.11 *Corrección* CONTIENE, Sección 2.5.20, Capítulo 4 Sección 5.2.5, 6.6.1, 6.8.6, 7.10.1, 8.11.1, 9.3, 11.2, APP 1.2, APP 3, APP 4
Jun., 2001	SH (NA)-080039-D	*Modelos adicionales* Q00JCPU, Q00CPU, Q01CPU *Adición* Sección 3.9, 11.2.1, 11.2.2, APP 1.3, APP 3.1, APP 3.2, APP 4.1, APP 4.2 *Corrección* CONTIENE, Sección 1.1, 5.3.8, 5.7.1, 6.1.5 ,6.5.2, 6.6.1, 6.8.1, 6.8.2, 6.8.4, 6.8.7, 6.8.8, 6.8.9, 7.1.2, 7.1.4, 7.1.6, 7.1.8, 7.2.1, 7.2.2, 7.2.3, 7.2.4, 7.4.2, 7.5.12, 7.6.6, 7.6.7, 7.6.9, 7.6.10, 7.7.1, 7.7.2, 7.7.3, 7.7.4, 7.9.3, 7.14.1, 9.4, 11.2.2, APP 1.2, APP 1.3, APP 2.1, APP2.1.4, APP 3.2, APP 4.2
Mar., 2002	SH (NA)-080039-E	*Modelos adicionales* Q12PHCPU, Q25PHCPU *Adición* Sección 11.2.3, APP 3.3, APP 4.3 *Corrección* CONTIENE, Sección 1.1, 1.2, 3.2.2, 3.6, 3.8, 6.6.1, Capítulo 9, Sección 9.10, APP 1.1
Dic., 2002	SH (NA)-080039-F	Revisión y reexaminación total para la compatibilidad con los modelos Básicos QCPU serie No. 04122
Mar., 2003	SH (NA)-080039-G	*Corrección* Sección 11.2.1, 11.2.2, 11.2.3, APP 1.3, APP 3.2, APP 4.2

Manual Versión Japonesa SH-080021-G

A - 2

Este manual no confiere derechos de propiedad o ninguna acción de otro tipo, tampoco confiere ninguna licencia de patentes. Mitsubishi Electric Corporation no se puede responsabilizar por ningún problema referente a los derechos industriales de propiedad que pueden ocurrir como resultado del uso del contenido de este manual.

© 2004 MITSUBISHI ELECTRIC CORPORATION

A - 3

INTRODUCCIÓN

Gracias por comprar Mitsubishi MELSEC-Q Series (Q mode) y MELSEC-QnA Series de Controladores Programables Lógicos.
Antes de utilizar este producto, por favor lea este manual cuidadosamente para familiarizarse con las funciones y tipos de Controladores Programables Lógicos Q Series (Q mode)/QnA Series que usted ha comprado, para poder asegurar un correcto uso.
Una copia de este manual debería ser entregada al usuario Final.

CONTIENE

1. DESCRIPCIÓN GENERAL	1 - 1 a 1 - 4

1.1 Manuales de programación ..... 1 - 1
1.2 Abreviaturas y numenclatura genérica ..... 1 - 4

2. TABLAS DE INSTRUCCIONES	2 - 1 a 2 - 53

2.1 Tipo de instrucciones ..... 2 - 1
2.2 Como leer las tablas de instrucciones ..... 2 - 3
2.3 Secuencia de instrucciones ..... 2 - 6
2.3.1 Instrucción de contacto ..... 2 - 6
2.3.2 Instrucciones de conexionado ..... 2 - 7
2.3.3 Instrucciones de salida ..... 2 - 8
2.3.4 Instrucciones de desplazamiento ..... 2 - 8
2.3.5 Instrucciones Control Master ..... 2 - 9
2.3.6 Instrucciones de fin ..... 2 - 9
2.3.7 Otras Instrucciones ..... 2 - 9
2.4 Instrucciones Básicas ..... 2 - 10
2.4.1 Instrucción de comparación ..... 2 - 10
2.4.2 Instrucción operación Aritmética ..... 2 - 17
2.4.3 Instrucciones conversión de datos ..... 2 - 22
2.4.4 Instrucciones transferencia de datos ..... 2 - 24
2.4.5 Instrucción ramificación de programa ..... 2 - 26
2.4.6 Instrucciones de control de ejecución de programa ..... 2 - 26
2.4.7 Instrucciones de refresco de E/S ..... 2 - 26
2.4.8 Otras instrucciones relevantes ..... 2 - 27
2.5 Instrucciones de aplicación ..... 2 - 28
2.5.1 Instrucciones operaciones lógicas ..... 2 - 28
2.5.2 Instrucciones de rotación ..... 2 - 31
2.5.3 Instrucciones de desplazamiento ..... 2 - 32
2.5.4 Instrucciones de procesamiento de bits ..... 2 - 33
2.5.5 Instrucciones de procesamiento de datos ..... 2 - 34
2.5.6 Instrucciones de creación de estructura ..... 2 - 36
2.5.7 Instrucciones de operación de tabla de datos ..... 2 - 37
2.5.8 Instrucciones de acceso a buffer de memoria ..... 2 - 38
2.5.9 Instrucciones de visualización ..... 2 - 38
2.5.10 Instrucciones de depuración de fallos y depuración ..... 2 - 39

2.5.11 Instrucciones de procesamiento de cadenas de caracteres ..... 2 - 40
2.5.12 Instrucciones de función especiales ..... 2 - 43
2.5.13 Instrucción de control de datos ..... 2 - 45
2.5.14 Instrucciones de conmutado ..... 2 - 46
2.5.15 Instrucciones de reloj (Reloj de Tiempo Real – RTC) ..... 2 - 47
2.5.16 Instrucciones de dispositivos de periféricos ..... 2 - 48
2.5.17 Instrucciones de control de programa ..... 2 - 48
2.5.18 Otras Instrucciones ..... 2 - 49
2.5.19 Instrucciones para comunicación de datos ..... 2 - 50
2.5.20 Instrucciones de QCPU ..... 2 - 52
2.5.21 Instrucciones de sistema redundantes (Para Q4ARCPU) ..... 2 - 53

3. CONFIGURACIÓN DE INSTRUCCIONES 3 - 1 a 3 - 35

3.1 Instrucciones de configuración ..... 3 - 1
3.2 Denominación de datos ..... 3 - 2
3.2.1 Marcas bit ..... 3 - 2
3.2.2 Registros de 16 bits ..... 3 - 3
3.2.3 Uso de datos de doble palabra (32 bits) ..... 3 - 6
3.2.4 Uso de datos numéricos reales ..... 3 - 9
3.2.5 Uso de datos de cadenas de caracteres ..... 3 - 11
3.3 Modificación de índice ..... 3 - 12
3.4 Designación indirecta ..... 3 - 16
3.5 Procesado de subconjunto ..... 3 - 19
3.6 Precauciones en programación (Errores de funcionamiento) ..... 3 - 21
3.7 Condiciones de ejecución de instrucciones ..... 3 - 24
3.8 Contando el número de pasos ..... 3 - 25
3.9 Funcionamiento cuando las instrucciones OUT, SET/RST, o PLS/PLF utilizan el mismo dispositivo 3 - 26
3.10 Precauciones para el uso de archivo de registros ..... 3 - 32
3.11 Precauciones para el uso de memoria RAM ..... 3 - 35

4. CONFIGURACIÓN DE LA SECUENCIA DE PROGRAMA Y EJECUCIÓN DE CONDICIONES 4 - 1 a 4 - 37

4.1 Secuencia de programa ..... 4 - 1
4.1.1 Programas de rutina principal ..... 4 - 3
4.1.2 Programas de subrutina ..... 4 - 4
4.1.3 Programas de interrupción ..... 4 - 5
4.2 Tipo de ejecución de programa ..... 4 - 12
4.2.1 Definición de programa de tipo de ejecución inicial ..... 4 - 16
4.2.2 Programa de ejecución por scan ..... 4 - 19
4.2.3 Programa de ejecución a baja velocidad ..... 4 - 21
4.2.4 Programa en Stand-by ..... 4 - 28
4.2.5 Programa de tipo de ejecución scan fijo ..... 4 - 35

5. ASIGNACIÓN DE E/S 5 - 1 a 5 - 15

5.1 Asignación de E/S ..... 5 - 1
5.1 Relación entre el número de puntos y los Slots de la Unidad Base de Extensión ..... 5 - 1
5.2 Instalación de las unidades base de extensión y configuración del número de puntos ..... 5 - 2
5.3 Asignación Unidad Base (Base Mode) ..... 5 - 3
5.4 ¿Cuáles son los números E/S? ..... 5 - 8
5.5 Concepto de Asignación de Número E/S ..... 5 - 9
5.5.1 Números E/S de la unidad base principal y la unidad base de extensión ..... 5 - 9
5.5.2 Estación Remota número E/S ..... 5 - 11
5.6 Asignación de E/S mediante Gx Developer ..... 5 - 12
5.6.1 Determinación de E/S mediante Gx Developer ..... 5 - 12
5.6.2 Concepto de Asignación de E/S mediante Gx Developer ..... 5 - 14

6MEMORIAS Y ARCHIVOS TRATADOS POR EL MODELO QCPU 6 - 1 a 6 - 30

6.1 Sobre la memoria del modelo QCPU de altas prestaciones ..... 6 - 3
6.2 Memoria de Programa ..... 6 - 7
6.3 A cerca de la ROM estándar ..... 6 - 9
6.4 A cerca de la RAM estándar ..... 6 - 10
6.5 Tarjeta de memoria ..... 6 - 12
6.6 Escritura en la ROM estándar o la tarjeta Flash ..... 6 - 14
6.6.1 Escritura de la ROM estándar o la tarjeta Flash mediante Gx Developer ..... 6 - 14
6.6.2 Escritura Automática en la ROM estándar (Auto descarga de toda la información de la tarjeta de memoria a la ROM estándar) ..... 6 - 16
6.7 Ejecución de programas ROM estándar de la tarjeta de memoria (ejecución de arranque) ..... 6 - 19
6.8 Configuración de archivo de programa ..... 6 - 22
6.9 Operación de archivos de Gx Developer y Precauciones de utilización ..... 6 - 24
6.9.1 Funcionamiento de archivo ..... 6 - 24
6.9.2 Precauciones de utilización de archivos ..... 6 - 26
6.9.3 Capacidad de archivo ..... 6 - 27
6.9.4 Capacidad de Memoria para archivos ..... 6 - 28

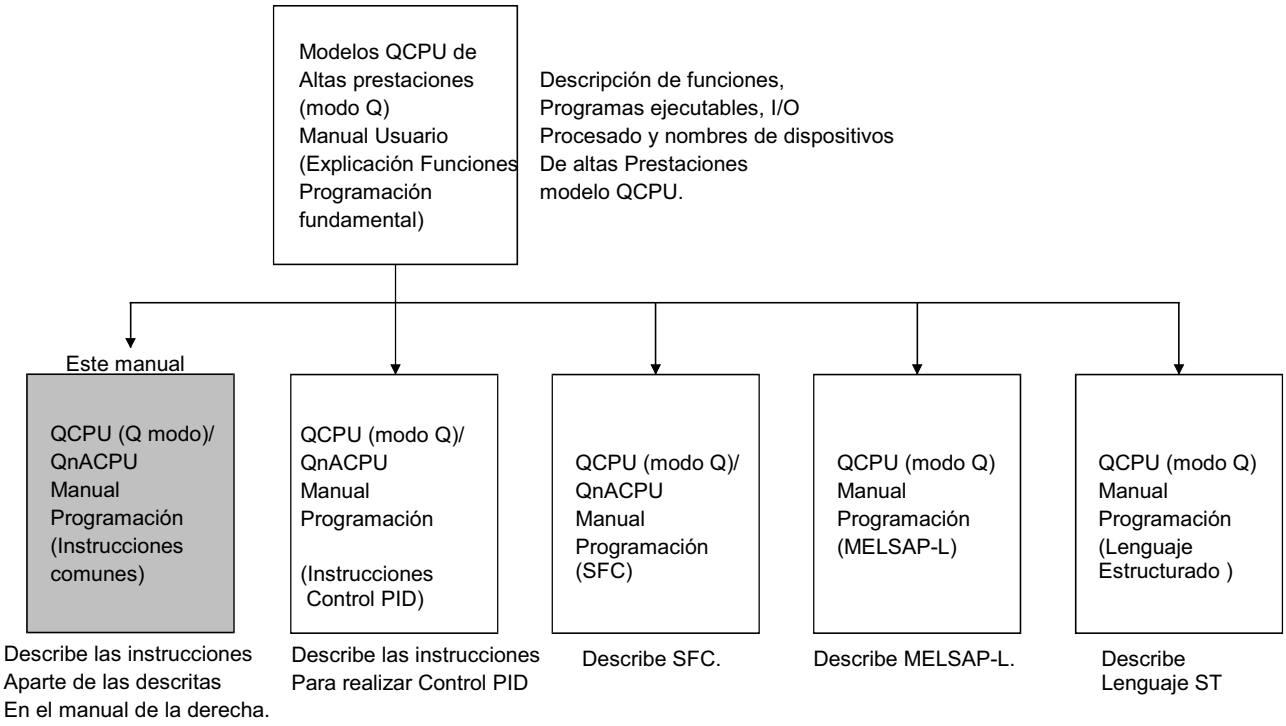
1. DESCRIPCIÓN GENERAL

Este manual describe las instrucciones comunes para el uso QCPU, QnACPU y Q2AS(H)CPU(S1) que son necesarias para poder programar correctamente QCPU (modo Q), QnACPU y Q2AS(H)CPU(S1).
Además de las Instrucciones comunes también se analizan instrucciones especiales para los módulos, como AJ71QC24 y AJ71PT32-S3, para AD57, para PID control, y para MELSAP3 y MELSAP-L.

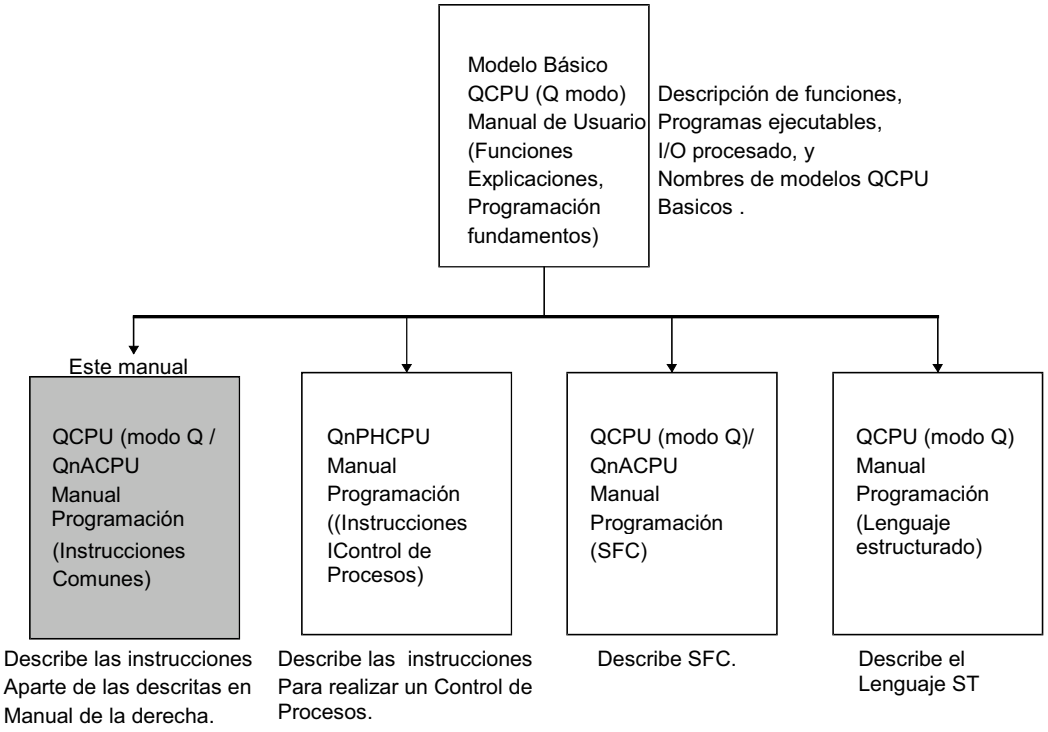
1.1 Relación Manuales de Programación

Antes de leer este manual, analice los programas, procesos I/O, y dispositivos que pueden ser usados con su módulo de CPU en el Manual de Usuario del Módulo CPU o bien en el Manual de Programación QnACPU (Fundamentos).

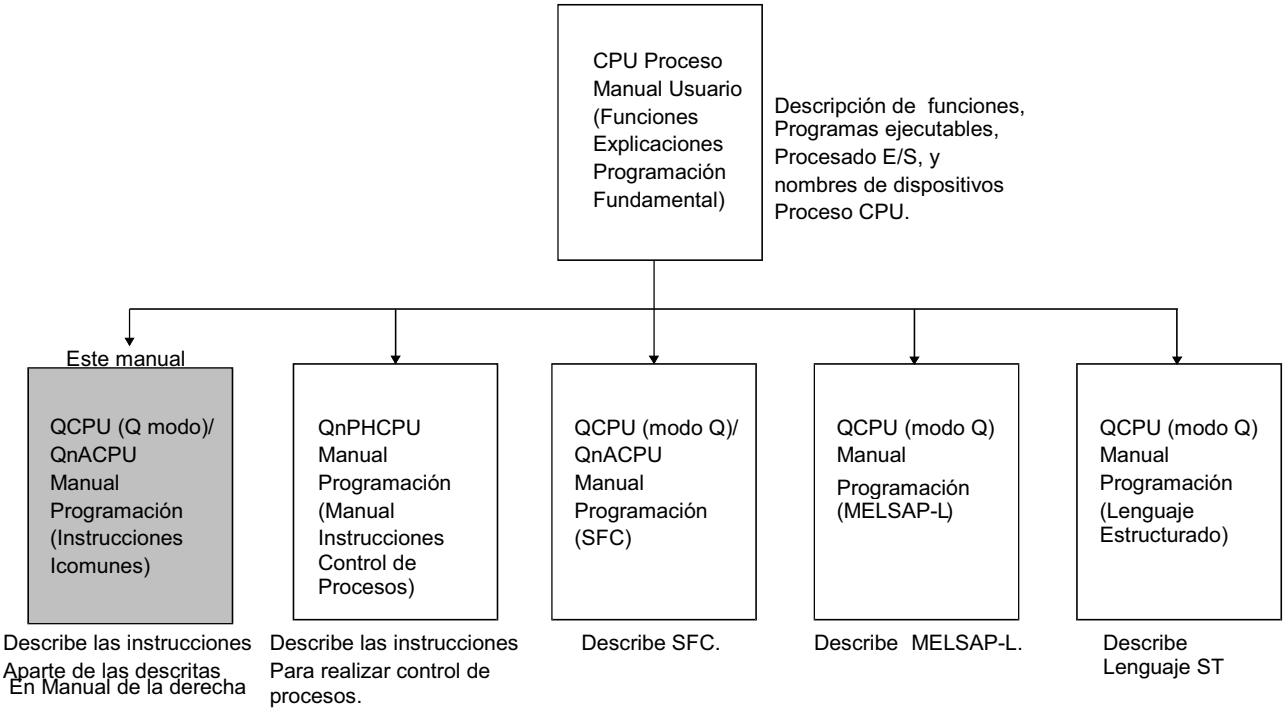
(1) Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU


Modelos QCPU de Altas prestaciones (modo Q) Manual Usuario (Explicación Funciones Programación fundamental)
Descripción de funciones, Programas ejecutables, I/O Procesado y nombres de dispositivos De altas Prestaciones modelo QCPU.
Este manual
QCPU (Q modo)/ QnACPU Manual Programación (Instrucciones comunes)
QCPU (modo Q)/ QnACPU Manual Programación (Instrucciones Control PID)
QCPU (modo Q)/ QnACPU Manual Programación (SFC)
QCPU (modo Q) Manual Programación (MELSAP-L)
QCPU (modo Q) Manual Programación (Lenguaje Estructurado)
Describe las instrucciones Aparte de las descritas En el manual de la derecha.
Describe las instrucciones Para realizar Control PID
Describe SFC.
Describe MELSAP-L.
Describe Lenguaje ST

(2) Q00JCPU, Q00CPU, Q01CPU

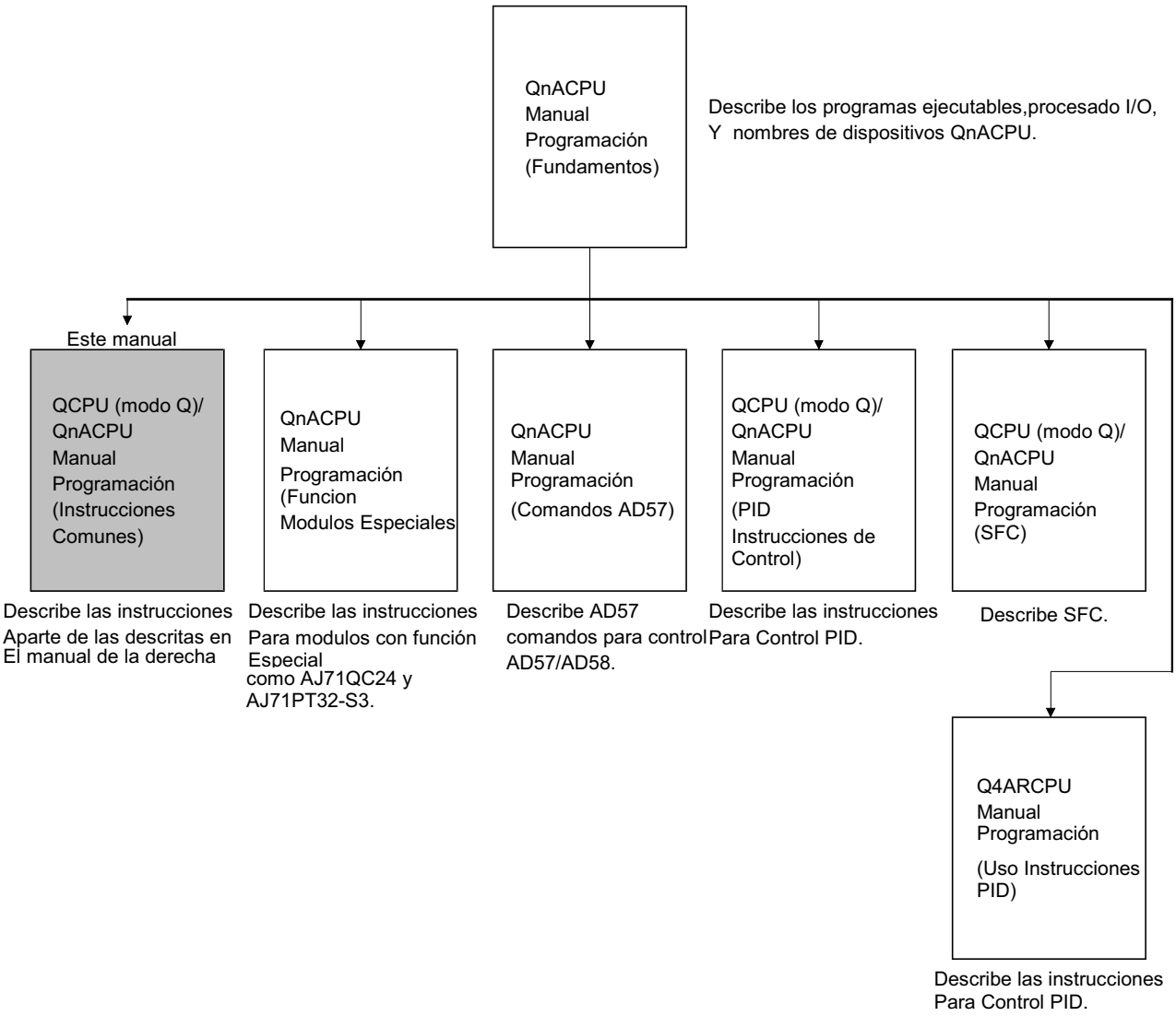

Modelo Básico QCPU (Q modo) Manual de Usuario (Funciones Explicaciones, Programación fundamentos)
Descripción de funciones, Programas ejecutables, I/O procesado, y Nombres de modelos QCPU Basicos .
Este manual
QCPU (modo Q / QnACPU Manual Programación (Instrucciones Comunes)
QnPHCPU Manual Programación ((Instrucciones IControl de Procesos)
QCPU (modo Q)/ QnACPU Manual Programación (SFC)
QCPU (modo Q) Manual Programación (Lenguaje estructurado)
Describe las instrucciones Aparte de las descritas en Manual de la derecha.
Describe las instrucciones Para realizar un Control de Procesos.
Describe SFC.
Describe el Lenguaje ST

(3) Q12PHCPU, Q25PHCPU


CPU Proceso Manual Usuario (Funciones Explicaciones Programación Fundamental)
Descripción de funciones, Programas ejecutables, Procesado E/S, y nombres de dispositivos Proceso CPU.
Este manual
QCPU (Q modo)/ QnACPU Manual Programación (Instrucciones Icomunes)
QnPHCPU Manual Programación (Manual Instrucciones Control de Procesos)
QCPU (modo Q)/ QnACPU Manual Programación (SFC)
QCPU (modo Q) Manual Programación (MELSAP-L)
QCPU (modo Q) Manual Programación (Lenguaje Estructurado)
Describe las instrucciones Aparte de las descritas En Manual de la derecha
Describe las instrucciones Para realizar control de procesos.
Describe SFC.
Describe MELSAP-L.
Describe Lenguaje ST

1 - 2

(4) Q2ACPU, Q3ACPU, Q4ACPU, Q4ARCPU, Q2AS(H)CPU



1.2 Abreviaturas y Nomenclatura Genérica

Los nombres de Módulos se abrevian a continuación

<table>
<tr><th>Modulo Tipo Nombre</th><th>Abreviatura</th><th>Abreviatura en Tablas</th><th>Nombre Genérico</th></tr>
<tr><td>Q00JCPU PLC CPU
Q00CPU PLC CPU
Q01CPU PLC CPU
Q02CPU PLC CPU
Q02HCPU PLC CPU
Q06HCPU PLC CPU
Q12HCPU PLC CPU
Q25HCPU PLC CPU
Q12PHCPU PLC CPU
Q25PHCPU PLC CPU</td><td>QCPU</td><td>——</td><td rowspan="8">CPU</td></tr>
<tr><td>Q00JCPU PLC CPU
Q00CPU PLC CPU
Q01CPU PLC CPU</td><td>Modelo QCPU Básico</td><td>Modelo QCPU Básico</td></tr>
<tr><td>Q02CPU PLC CPU
Q02HCPU PLC CPU
Q06HCPU PLC CPU
Q12HCPU PLC CPU
Q25HCPU PLC CPU</td><td>Modelo QCPU de Altas Prestaciones</td><td>Modelo QCPU de Altas Prestaciones</td></tr>
<tr><td>Q12PHCPU PLC CPU
Q25PHCPU PLC CPU</td><td>CPU Proceso</td><td>CPU Proceso</td></tr>
<tr><td>Q2ACPU(S1) PLC CPU
Q3ACPU PLC CPU
Q4ACPU PLC CPU</td><td>QnACPU</td><td>QnA</td></tr>
<tr><td>Q2ASCPU(S1) PLC CPU
Q2ASHCPU(S1) PLC CPU</td><td>Q2ASCPU</td><td>Q2AS</td></tr>
<tr><td>Q4ARCPU PLC CPU</td><td>Q4ARCPU</td><td>Q4AR</td></tr>
<tr><td>Sistema de red MELSECNET/H (MELSECNET/H modo)</td><td rowspan="2">MELSECNET/H</td><td rowspan="3">MELSECNET/10(H)</td><td rowspan="3">——</td></tr>
<tr><td>Sistema de red MELSECNET/H (MELSECNET/10 modo)</td></tr>
<tr><td>Sistema de red MELSECNET/10</td><td>MELSECNET/10</td></tr>
<tr><td>Modulo Interfaz Ethernet</td><td>Modulo Interfaz Ethernet</td><td>Modulo Interfaz Ethernet</td><td>——</td></tr>
<tr><td>Sistema de enlace de control y comunicación maestro/ esclavo</td><td>modulo CC-Link</td><td>modulo CC-Link</td><td>——</td></tr>
</table>

2. TABLAS DE INSTRUCCIONES

2.1 Tipos de Instrucciones

La mayoría de tipos de instrucciones de las diferentes CPU consisten en una secuencia de instrucciones tales como instrucciones básicas, instrucciones de aplicación, instrucciones de enlace de datos, instrucciones referentes a QCPU e instrucciones de sistemas redundantes. Todo este tipo de instrucciones está en la lista de la tabla adjunta 2.1.

Tabla 2.1 Instrucciones

Tipo de Instrucción		Explicación	Capítulo de Referencia
Instrucciones de secuencia	Instrucciones de contacto	Operación de inicio, conexión en serie y en paralelo	5
	Instrucciones de conexión	Bloques de conexión Ladder, creación de pulsos desde resultados de operación, almacenamiento lectura de resultados de operación	
	Instrucciones de Salida	Bit de dispositivo de salida, pulso de salida, salida reversible	
	Instrucciones de desplazamiento	Bit de desplazamiento	
	Instrucción de Master control	Master control	
	Instrucción de fin	Final de Programa	
	Otras instrucciones	Parada del programa, instrucciones como no-operación la cual nos se debe incluir en las categorías superiores	
Instrucciones básicas	Instrucción de comparación	Comparaciones tales como =, >, <	6
	Instrucciones aritméticas	Suma, resta, multiplicación o división en Binario o BCD	
	Instrucciones de conversión BCD ←→ Binario	Conversión de BCD a Binario y de Binario a BCD	
	Instrucciones de transferencia de datos	Datos destinados para su transmisión	
	Instrucciones de bifurcación de programa	Salto de programa	
	Instrucción de habilitación del control del programa	Habilita o deshabilita las interrupciones de programa	
	Refresco de E/S	Refresco inicial de puesta en marcha	
	Otras instrucciones a destacar	Instrucciones para contaje Incremento / Decremento, temporizadores, funciones especiales para temporizadores, tablas cortas de direccionamiento de control, etc.	
Instrucciones de Aplicación	Instrucciones de operaciones lógicas	Operaciones lógicas tales como suma lógica, producto lógico, etc.	7
	Instrucciones de rotación	Rotaciones de datos específicos	
	Instrucciones desplazamiento	Desplazamiento de datos específicos	
	Instrucciones de procesamiento de bits	Activación y desactivación de bits, bit de testeo, batch test de dispositivo de bits	
	Instrucciones de procesamiento de datos	Datos de 16-bit de búsqueda, procesamiento de datos tales como decodificación y decodificación	
	Instrucciones de creación de estructuras	Operaciones redundantes, programación de llamada a subrutinas, modificación de índices en unidades ladder	
	Tablas de instrucciones de operación	Lectura / Escritura de tablas FIFO	
	Instrucciones de acceso a la interfaz de memoria	Datos de Lectura / escritura para módulos especiales de función	
	Instrucciones de Display	Código de impresión ASCII, visualización de carácter Leed etc.	
	Depuración y diagnóstico de instrucciones de operación	Chequeo, estado de chequeo, señal de muestreo, señal de programa	
	Instrucciones de procesamiento de cadenas de caracteres	Conversión entre BIN / BCD y ASCII; conversión entre BIN y cadenas de caracters; conversión entre datos decimales en coma flotante y cadenas de caracteres, cadenas de caracteres en proceso, etc.	
	Instrucciones funciones especiales	Funciones trigonométricas, conversión de ángulos a radianes, operaciones exponenciales, logaritmos automáticos, raíces cuadradas	
	Instrucciones de control de datos	Subida y bajada de límites de control, control de zona muerta, zonas de control	
	Instrucciones de conexión	Cambio de número de bloque de fichero de registros. Designación de fichero de registros y comentarios	
	Instrucciones de reloj	Lectura / Escritura del año, mes, día, hora, minuto, Segundo y día de la semana conversión de sentencia temporal (hora, minutos, segundos) y segundos	
	Instrucciones de dispositivos periféricos	E/S conectadas a periféricos	
	Instrucciones de Programa	Instrucciones para cambiar las condiciones de ejecución de programa	

	Otras instrucciones	Instrucciones que no se encuentran en las categorías superiores tales como perro guardián (whatchdog timer) instrucciones de reset e instrucciones de conteo de reloj	
Instrucciones de intercambio de datos a redes	Instrucciones de refresco de comunicación	Refresco de datos de redes	8
	Instrucciones para la comunicación de QnA	Lectura / Escritura de datos de otras estaciones; señales de datos de transmisión hacia otras estaciones; procesamiento de peticiones hacia otras estaciones	
	Instrucciones de comunicación compatible para serie A	Lectura / escritura de datos desde otras estaciones; señales de transmisión de datos a otras estaciones; procesamiento de encuestas para otras estaciones	
	Instrucciones de enrutamiento de órdenes de lectura/ escritura	Lecturas, escrituras y registros de enrutamiento de información	
Instrucciones de QCPU	Instrucciones para QCPU	Módulo de lectura de información; señalización de set / reset; lectura/escritura de datos binarios; carga/descarga + descarga del programa desde tarjeta de memoria	9
Instrucciones de sistemas redundantes	Instrucciones para Q4ARCPU	Operación de modo de posición durante la inicialización de la CPU; Operación de modo de posición de instrucciones durante la conmutación de la CPU; seguimiento de datos; refresco en lotes de la memoria de datos	10

2.2 Cómo leer tablas de instrucciones

Las tablas de instrucciones encontradas desde el apartado 2.3 hasta el 2.6 han sido elaboradas de acuerdo con el siguiente formato:

Tabla 2.2 Como leer tablas de instrucciones

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condiciones de ejecución	Número de pasos	Subprograma	Ver la descripción
Operaciones de suma y resta BIN 16-bit	+	+ S D	• (D)+(S)→(D)		3	●	6-16
	+P	+P S D					
	+	+ S1 S2 D	• (S1)+(S2)→(D)		4	●	6-18
	+P	+P S1 S2 D					

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

Descripción

①.....Instrucciones clasificadas de acuerdo con su aplicación.

②.....Indica el símbolo de la instrucción d el programa.

El código de la instrucción está constituido por 16 bits de instrucciones. Las siguientes notaciones se usarán para instrucciones en modo de 32 bits, instrucciones ejecutadas sólo en el flanco ascendente de OFF a ON, instrucciones de números reales e instrucciones de cadena de caracteres:

- Instrucciones de 32 bits ..... La letra "D" se añade a la primera línea de la instrucción

Ejemplo + → D+

Instrucción 16-bit Instrucción 32-bit

- Instrucciones solamente ejecutadas en el flanco ascendente de OFF a ON ..... La letra "P" es añadida a la última línea de la instrucción

Ejemplo + → +P

Instrucciones Ejecutadas en estado ON

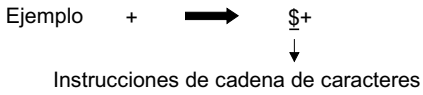
Instrucciones ejecutadas sólamente Durante la conmutación de OFF a ON

- Instrucciones para números reales ..... La letra "E" es añadida a la primera línea de la instrucción

Ejemplo + → E+

Instrucciones para números reales

- Instrucciones referentes a cadenas de caracteres
  ..... El signo de dólar “$”es añadido a la primera línea de la instrucción



③.....Muestra del diagrama de símbolos en programación ladder

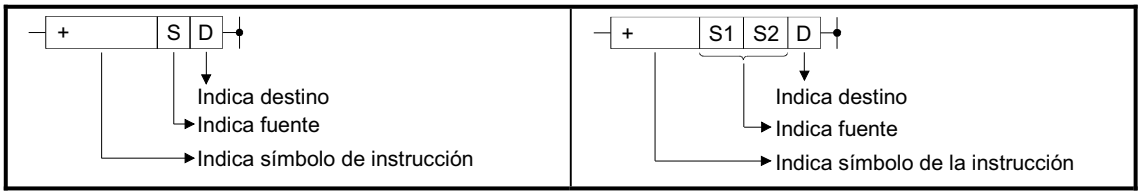


Fig. 2.1 Muestra del diagrama de símbolos en programación ladder

Destino ..... Indica dónde será enviado el dato después de la operación
Fuente ..... Indica el dato previo a la operación

④.....Indica el tipo de procesamiento que ejecutan las instrucciones individuales

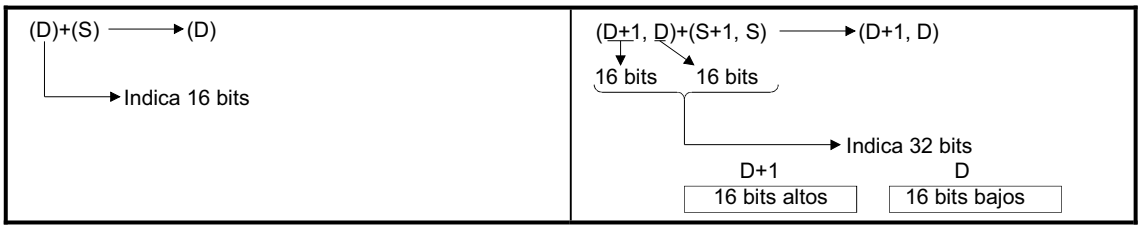


Fig. 2.2 Tipo de procesamiento ejecutado por instrucciones individuales

⑤.....Los detalles de las condiciones para la ejecución de instrucciones individuales son los siguientes:

Símbolo	Condición de ejecución
Sin símbolo	Instrucción ejecutada bajo condiciones normales sin considerar el estado ON / OFF de condiciones previas para la instrucción. Si la precondición es OFF, la instrucción procesará OFF.
	Ejecutado durante ON; la instrucción es ejecutada sólo mientras la precondición es ON. Si la precondición es OFF, la instrucción no es ejecutada y no habrá proceso.
	Ejecutado una vez está a ON; instrucción ejecutada sólo en el flanco ascendente cuando la precondición va de OFF a ON. Seguidamente, la instrucción no será ejecutada e incluso no habrá procesamiento si la condición continua siendo ON.
	Ejecutado cuando está a OFF; instrucción sólo ejecutada mientras la precondición es OFF. Si la precondición es ON, la instrucción no es ejecutada y no hay procesamiento.
	Ejecutado una vez está a OFF; instrucción ejecutada solo cuando hay flanco descendente

⑥.....Indica el número básico de pasos para instrucciones individuales.
Obsérvese en la sección 3.8 la descripción del número de pasos.

⑦ La marca "●" indica instrucciones en las que el procesamiento de subprograma es posible.

⑧.....Indica el número de página donde las instrucciones individuales están explicadas.

2.3 Secuencia de instrucciones

2.3.1 Instrucciones de contacto.

Tabla 2.3 Instrucciones de contacto

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Contacto	LD		• Inicio de operación lógica (conmuta un contacto normalmente abierto)		＊1	● ＊3	5-2
	LDI		• Inicio de operación lógica negada (conmuta un contacto normalmente cerrado)				
	AND		• Producto lógico (conexión en serie de un contacto normalmente abierto)				
	ANI		• Producto lógico negado (conexión en serie de un contacto normalmente cerrado)				
	OR		• Suma lógica (conexión en paralelo de un contacto normalmente abierto)				
	ORI		• Suma lógica negada (conexión en paralelo de un contacto normalmente abierto)				
	LDP		• Ejecuta un pulso por flanco ascendente		＊2	● ＊3	5-5
	LDF		• Ejecuta un pulso por flanco descendente				
	ANDP		• Conexión serie mediante pulso de flanco ascendente				
	ANDF		• Conexión serie mediante pulso de flanco descendente				
	ORP		• Conexión paralelo mediante pulso de flanco ascendente				
	ORF		• Conexión paralelo mediante pulso de flanco descendente				

Comentarios

1) ＊1: El número de pasos puede variar despendiendo de los dispositivos que son usados.

Dispositivo	Número de pasos
Dispositivo interno, registros (R0 a R32767)	1
Acceso directo a entrada (DX)	2
Otros dispositivos	3

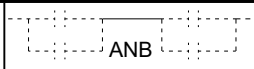
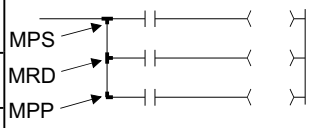
2) ＊2: El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utiliza.

Dispositivo	Número de pasos	
	QCPU	QnACPU
Dispositivo interno, registros (R0 a R32767)	1	2
Acceso directo a entrada (DX)	2	2
Otros dispositivos	3	3

3) ＊3: El subprograma se ejecuta únicamente con el modelo QCPU.

2.3.2 Instrucciones de conexionado.

Tabla 2.4 Instrucciones de conexionado

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Conexión	ANB		• Función AND entre bloques lógicos (Conexión serie entre bloques lógicos)		1	—	5-7
	ORB		• Función OR entre bloques lógicos (Conexión serie entre bloques lógicos)				
	MPS		• Memoria de almacenamiento de resultados de operación		1	—	5-9
	MRD		• Lectura del resultado de operación almacenada mediante la instrucción MPS				
	MPP		• Lectura y reset de operación almacenado				
	INV		• Inversión del resultado de la operación de inversión.		1	—	5-13
	MEP		• Conversión del resultado de la operación por pulso de flanco ascendente.		1	—	5-14
	MEF		• Conversión del resultado de la operación por pulso de flanco descendente.				
	EGP		• Conversión del resultado de la operación por pulso de flanco ascendente. (Almacenado en Vn)		1	—	5-16
	EGF		• Conversión del resultado de la operación por pulso de flanco descendente. (Almacenado en Vn)		∗1		

Comentarios

∗1: El número de pasos puede variar despendiendo de los dispositivos que son usados.

Componente	Número de pasos
Modelo QCPU de altas prestaciones Procesamiento CPU de procesos QnACPU	1
Modelo QCPU básico	2

2.3.3 Instrucciones de salida.

Tabla 2.5 Instrucciones de salida.

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Salida	OUT	—( )—	• Dispositivo de salida		*1	—	5-18
	SET	— SET D —	• Puesta a on de dispositivo (set)	*2	*1	—	5-28 5-32
	RST	— RST D —	• Puesta a off de dispositivo	*2	*1	—	5-30 5-32
	PLS	— PLS D —	• Genera un pulso de 1 ciclo de programa por flanco ascendente de una señal de entrada		2	—	5-34
	PLF	— PLF D —	• Genera un pulso de 1 ciclo de programa en el flanco descendente de una señal de entrada				
	FF	— FF D —	• Inversión de la señal de salida		2	—	5-36
	DELTA	— DELTA D —	• Conversión de pulso de salida directa		2	—	5-38
	DELTAP	— DELTAP D —					

Comentarios

1) *1: El número de pasos puede variar despendiendo de los dispositivos que son usados. Véase páginas de descripción de instrucciones individuales en referencia a número de pasos.
2) *2: La ejecución de ⌐↑ solamente se aplica cuando el anunciador (F) se encuentra en uso.

2.3.4 Instrucciones de desplazamiento.

Tabla 2.6 Instrucciones de desplazamiento

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Desplaza_ miento	SFT	— SFT D —	• Desplazamiento de 1 bit de dispositivo		2	—	5-40
	SFTP	— SFTP D —					

2.3.5 Instrucciones de Control Master

Tabla 2.7 Instrucciones de Control Master

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Master control	MC	MC n D	• Inicia master control		2	—	5-42
	MCR	MCR n	• Resetea master control		1		

2.3.6 Instrucciones de fin.

Tabla 2.8 Instrucciones de fin

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Fin de programa	FEND	FEND	• Final del programa principal		1	—	5-46
	END	END	• Final de la secuencia de programa				5-48

2.3.7 Otras instrucciones.

Tabla 2.9 Otras instrucciones

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Stop	STOP	STOP	• Acaba la secuencia de programa después de que la condición de entrada ha sido encontrada. • Se ejecuta una secuencia de programa normalmente mediante el interruptor RUN / STOP en la posición de RUN		1	—	5-50
Ignorado	NOP	—	• Ignorado (Por borrado de programa o espacio)		1	—	5-52
	NOPLF	NOPLF	• Ignorado (Para cambiar páginas durante impresión)				
	PAGE	PAGE n	• Ignorado (Los subprogramas serán controlados desde el paso 0 de la página n)				

2.4 Instrucciones básicas.

2.4.1 Instrucciones de comparación.

Tabla 2.10 Instrucciones de comparación

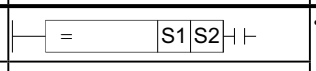
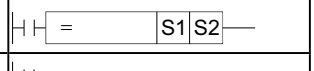
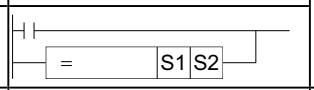
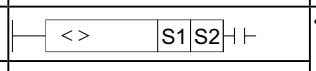
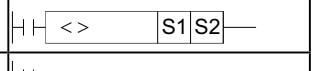
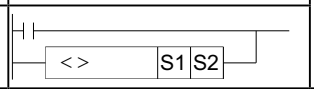
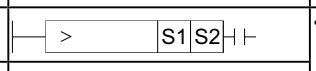
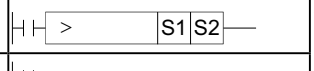
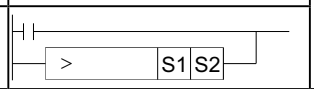
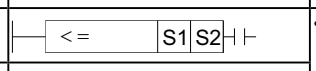
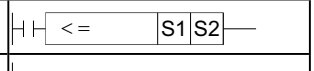
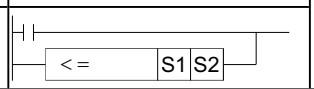
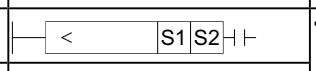
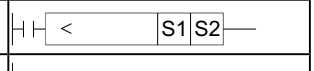
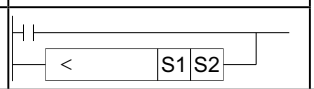
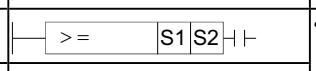
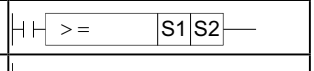
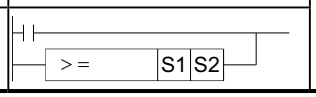
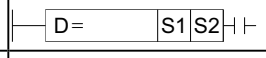
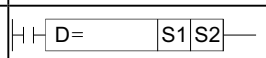
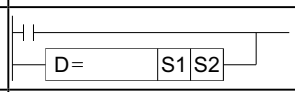
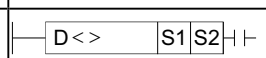
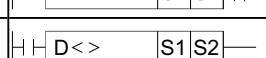
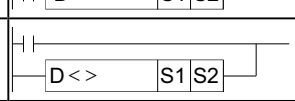
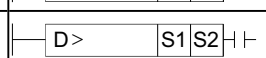
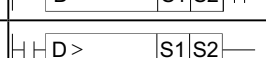
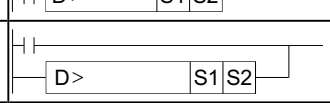
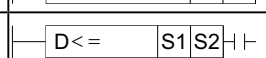
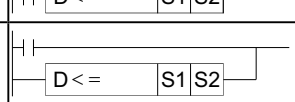
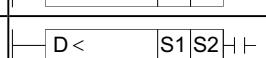
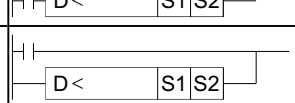
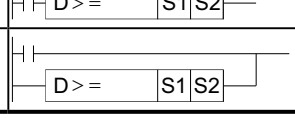
Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Comparaciones de datos de 16 bits	LD=		• Estado de conducción cuando (S1) = (S2) • Estado de no conducción cuando (S1) ≠ (S2)		3	●	6-2
	AND=						
	OR=						
	LD<>		• Estado de conducción cuando (S1) ≠ (S2) • Estado de no conducción cuando (S1) = (S2)		3	●	6-2
	AND<>						
	OR<>						
	LD>		• Estado de conducción cuando (S1) > (S2) • Estado de no conducción cuando (S1) ≤ (S2)		3	●	6-2
	AND>						
	OR>						
	LD<=		• Estado de conducción cuando (S1) ≤ (S2) • Estado de no conducción cuando (S1) > (S2)		3	●	6-2
	AND<=						
	OR<=						
	LD<		• Estado de conducción cuando (S1) < (S2) • Estado de no conducción cuando (S1) ≥ (S2)		3	●	6-2
	AND<						
	OR<						
	LD>=		• Estado de conducción cuando (S1) ≥ (S2) • Estado de no conducción cuando (S1) < (S2)		3	●	6-2
	AND>=						
	OR>=						

Tabla 2.10 Instrucciones de comparación (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Comparaciones de datos de 32 bits	LDD=		• Estado de conducción cuando (S1+1, S1) = (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≠ (S2+1, S2)		＊1	●	6-4
	ANDD=						
	ORD=						
	LDD<>		• Estado de conducción cuando (S1+1, S1) ≠ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) = (S2+1,S2)		＊1	●	6-4
	ANDD<>						
	ORD<>						
	LDD>		• Estado de conducción cuando (S1+1, S1) > (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≤ (S2+1, S2)		＊1	●	6-4
	ANDD>						
	ORD>						
	LDD<=		• Estado de conducción cuando (S1+1, S1) ≤ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) > (S2+1, S2)		＊1	●	6-4
	ANDD<=						
	ORD<=						
	LDD<		• Estado de conducción cuando (S1+1, S1) <(S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≥ (S2+1, S2)		＊1	●	6-4
	ANDD<						
	ORD<						
	LDD>=		• Estado de conducción cuando (S1+1, S1) ≥ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) < (S2+1, S2)		＊1	●	6-4
	ANDD>=						
	ORD>=						

Comentarios

∗1 : El número de pasos puede variar despendiendo de los dispositivos y tipos de CPU que son usados.

Componente	Número de pasos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU de procesos	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	Nota 1) 5
	(2) Cuando se usan otros dispositivos (1)	Nota 2) 3
Modelo básico QCPU QnCPU	3	Nota 2)

Nota 1:Para el modelo QCPU o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

Nota 2:El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

Tabla 2.10 Instrucciones de operación de comparación (Continuación)

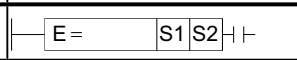
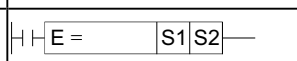
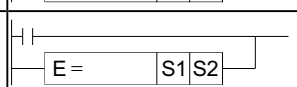
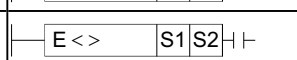
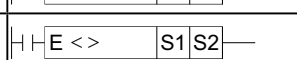
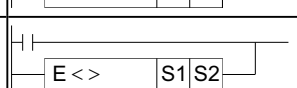
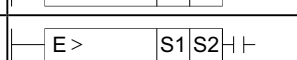
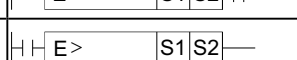
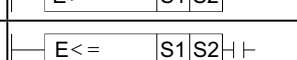
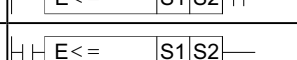
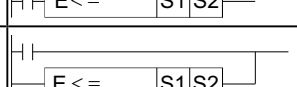
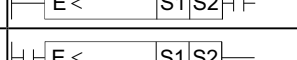
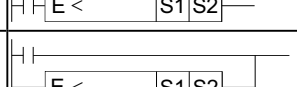
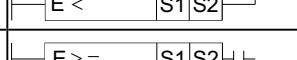
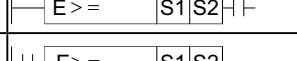
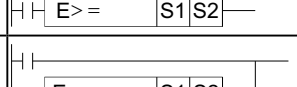
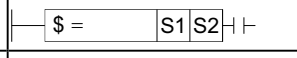
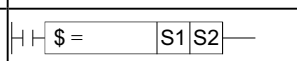
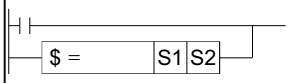
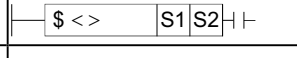
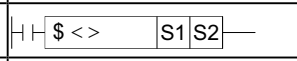
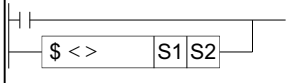
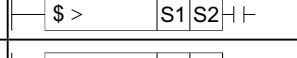
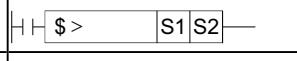
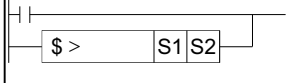
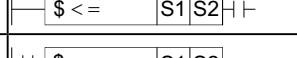
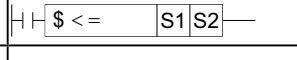
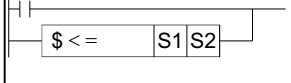
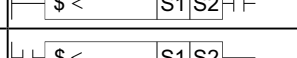
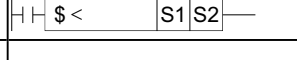
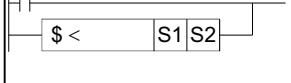
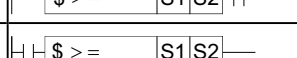
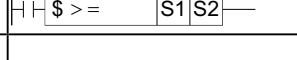
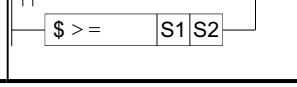
Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Comparación de datos de números reales	LDE=		• Estado de conducción cuando (S1+1, S1) = (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≠ (S2+1, S2)		3	—	6-6
	ANDE=						
	ORE=						
	LDE<>		• Estado de conducción cuando (S1+1, S1) ≠ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) = (S2+1, S2)		3	—	6-6
	ANDE<>						
	ORE<>						
	LDE>		• Estado de conducción cuando (S1+1, S1) > (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≤ (S2+1, S2)		3	—	6-6
	ANDE>						
	ORE>						
	LDE<=		• Estado de conducción cuando (S1+1, S1) ≤ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) > (S2+1, S2)		3	—	6-6
	ANDE<=						
	ORE<=						
	LDE<		• Estado de conducción cuando (S1+1, S1) < (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) ≥ (S2+1, S2)		3	—	6-6
	ANDE<						
	ORE<						
	LDE>=		• Estado de conducción cuando (S1+1, S1) ≥ (S2+1, S2) • Estado de no conducción cuando (S1+1, S1) < (S2+1, S2)		3	—	6-6
	ANDE>=						
	ORE>=						

Tabla 2.10 Instrucciones de comparación (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Comparación de datos de cadenas de caracteres	LD$=		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de conducción cuando (cadena de caracteres S1) = (cadena de caracteres S2) • Estado de no conducción (cadena de caracteres S1) ≠ (cadena de caracteres S2)		3	—	6-8
	AND$=						
	OR$=						
	LD$<>		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de conducción cuando (cadena de caracteres S1) ≠ (cadena de caracteres S2) • Estado de no conducción (cadena de caracteres S1) = (cadena de caracteres S2)		3	—	6-8
	AND$<>						
	OR$<>						
	LD$>		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de conducción cuando (cadena de caracteres S1) >(cadena de caracteres S2) • Estado de no conducción (cadena de caracteres S1) ≤ (cadena de caracteres S2)		3	—	6-8
	AND$>						
	OR$>						
	LD$<=		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de conducción cuando (cadena de caracteres S1) ≤ (cadena de caracteres S2) • Estado de no conducción (cadena de caracteres S1)>(cadena de caracteres S2)		3	—	6-8
	AND$<=						
	OR$<=						
	LD$<		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de conducción cuando (cadena de caracteres S1) < (cadena de caracteres S2) • Estado no conductivo cuando (cadena de caracteres S1) ≥ (cadena de caracteres S2)		3	—	6-8
	AND$<						
	OR$<						
	LD$>=		• Compara una cadena de caracteres S1 con la cadena de caracteres S2 cada carácter al mismo tiempo. ＊ • Estado de no conducción (cadena de caracteres S1) ≥ (cadena de caracteres S2) • Estado de no conducción cuando (cadena de caracteres S1) < (cadena de caracteres S2)		3	—	6-8
	AND$>=						
	OR$>=						

Comentarios

1) ＊ : Las condiciones bajo las cuales la comparación de cadenas de caracteres puede hacerse son:

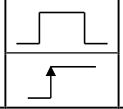
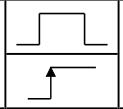
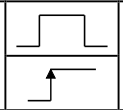
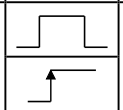
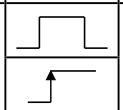
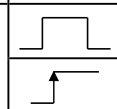
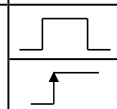
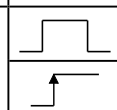
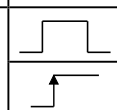
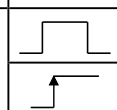
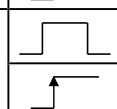
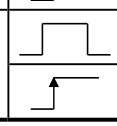
- Igualación: Todas las cadenas de caracteres deben coincidir.
- Cadena larga: Si las cadenas de caracteres son diferentes, determina la cadena con el mayor número de códigos de caracteres.
  Si la longitud de las cadenas de caracteres es diferente, determina la mayor cadena de caracteres.
- Cadena corta: Si las cadenas de caracteres son diferentes, determina la cadena con el menor número de códigos de caracteres.
  Si la longitud de las cadenas de caracteres es diferente, determina la menor cadena de caracteres.

Tabla 2.10 Instrucciones de operación de comparación (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Comparacion de bloques de datos	BKCMP=	BKCMP = S1 S2 D n	• Compara n puntos de datos de S1 con n puntos de datos de S2 en unidades de 1 palabra, y almacena el resultado de la comparación a los n puntos del dispositivo bit designado por (D).		5	—	6-12
	BKCMP<>	BKCMP < > S1 S2 D n					
	BKCMP>	BKCMP > S1 S2 D n					
	BKCMP<=	BKCMP < = S1 S2 D n					
	BKCMP<	BKCMP < S1 S2 D n					
	BKCMP>=	BKCMP > = S1 S2 D n					
	BKCMP=P	BKCMP = P S1 S2 D n					
	BKCMP< >P	BKCMP< > P S1 S2 D n					
	BKCMP>P	BKCMP > P S1 S2 D n					
	BKCMP<=P	BKCMP < = P S1 S2 D n					
	BKCMP<P	BKCMP < P S1 S2 D n					
	BKCMP>=P	BKCMP > = P S1 S2 D n					

2.4.2 Instrucciones aritméticas.

Tabla 2.11 Instrucciones aritméticas.

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Operaciones de suma y resta con números binarios de 16 bits	+	+ S D	• (D)+(S) → (D)		3	●	6-16
	+P	+P S D					
	+	+ S1 S2 D	• (S1)+(S2) → (D)		4	●	6-18
	+P	+P S1 S2 D					
	-	− S D	• (D) - (S) → (D)		3	●	6-16
	-P	−P S D					
	-	− S1 S2 D	• (S1) - (S2) → (D)		4	●	6-18
	-P	−P S1 S2 D					
Operaciones de suma y resta con números binarios de 32 bits	D+	D+ S D	• (D+1, D)+(S+1, S) → (D+1, D)		＊1	●	6-20
	D+P	D+P S D					
	D+	D+ S1 S2 D	• (S1+1, S1)+(S2+1, S2) → (D+1, D)		＊2	●	6-22
	D+P	D+P S1 S2 D					
	D-	D− S D	• (D+1, D)-(S+1, S) → (D+1, D)		＊1	●	6-20
	D-P	D−P S D					
	D-	D− S1 S2 D	• (S1+1, S1)-(S2+1, S2) → (D+1, D)		＊2	●	6-22
	D-P	D−P S1 S2 D					
Operaciones de multiplicación y división con números binarios de 16 bits	＊	＊ S1 S2 D	• (S1) ＊ (S2) → (D+1, D)		＊3	●	6-24
	＊P	＊P S1 S2 D					
	/	/ S1 S2 D	• (S1)/(S2) → Cociente (D), Resto (D+1)		4	●	6-24
	/P	/P S1 S2 D					
Operaciones de multiplicación y división con números binarios de 32 bits	D＊	D＊ S1 S2 D	• (S1+1, S1) ＊ (S2+1, S2) → (D+3, D+2, D+1, D)		4	●	6-26
	D＊P	D＊P S1 S2 D					
	D/	D/ S1 S2 D	• (S1+1, S1)/(S2+1, S2) → Cociente (D+1, D), Resto (D+3, D+2)		4	●	6-26
	D/P	D/P S1 S2 D					

Comentarios

1) ∗1: El Número de pasos puede variar según el dispositivo y tipo de modulo de CPU que sé este utilizando.

Componente	Número de pasos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	Nota 1) 5
	(2) En caso de utilización de otros dispositivos a parte de (1)	Nota 2) 3
Modelos básicos QCPU QnCPU	3	Nota 2)

Nota 1:Para el modelo QCPU o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

Nota 2:El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

2) ∗2:El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	Nota 1) 6
	(2) En caso de utilización de otros dispositivos a parte de (1)	Nota 2) 4
Modelos básicos QCPU QnCPU	4	Nota 2)

Nota 1:Para el modelo QCPU o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

Nota 2:El número de pasos puede incrementar debido a las condiciones descritas en la sección 3.8.

3) ∗3:El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos	
QCPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	3
	En caso de utilización de otros dispositivos a parte de los anteriores	Nota 1) 4
QnACPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	4
	En caso de utilización de otros dispositivos a parte de los anteriores	Nota 1) 4

Nota 1: El número de pasos puede incrementar debido a las condiciones descritas en la sección 3.8.

Tabla 2.11 Instrucciones de operaciones aritméticas (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Operaciones de suma y resta de 4 dígitos en BCD	B+	B+ S D	• (D)+(S) → (D)		3	●	6-28
	B+P	B+P S D					
	B+	B+ S1 S2 D	• (S1)+(S2) → (D)		4	—	6-30
	B+P	B+P S1 S2 D					
	B-	B− S D	• (D)-(S) → (D)		3	●	6-28
	B-P	B−P S D					
	B-	B− S1 S2 D	• (S1)-(S2) → (D)		4	—	6-30
	B-P	B−P S1 S2 D					
Operaciones de suma y resta de 8 dígitos en BCD	DB+	DB+ S D	• (D+1, D)+(S+1, S) → (D+1, D)		3	—	6-32
	DB+P	DB+P S D					
	DB+	DB+ S1 S2 D	• (S1+1, S1)+(S2+1, S2) → (D+1, D)		4	—	6-34
	DB+P	DB+P S1 S2 D					
	DB-	DB− S D	• (D+1, D)-(S+1, S) → (D+1, D)		3	—	6-32
	DB-P	DB−P S D					
	DB-	DB− S1 S2 D	• (S1+1, S1)-(S2+1, S2) → (D+1, D)		4	—	6-34
	DB-P	DB−P S1 S2 D					
Operaciones de producto y división de 4 dígitos en BCD	B*	B* S1 S2 D	• (S1) * (S2) → (D+1, D)		4	●	6-36
	B*P	B*P S1 S2 D					
	B/	B/ S1 S2 D	• (S1)/(S2) → Cociente (D), Resto (D+1)		4	●	6-36
	B/P	B/P S1 S2 D					
Operaciones de producto y división de 8 dígitos en BCD	DB*	DB* S1 S2 D	• (S1+1, S1) * (S2+1, S2) → (D+3, D+2, D+1, D)		4	—	6-38
	DB*P	DB*P S1 S2 D					
	DB/	DB/ S1 S2 D	• (S1+1, S1)/(S2+1, S2) → Cociente (D+1, D), Resto (D+3, D+2)		4	●	6-38
	DB/P	DB/P S1 S2 D					

Tabla 2.11 Instrucciones aritméticas (Continuación)

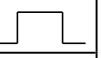
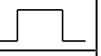
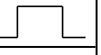
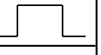
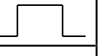
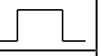
Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Operaciones de suma y resta de datos decimales en coma flotante.	E+	E+ S D	• (D+1, D)+(S+1, S) → (D+1, D)		3	—	6-40
	E+P	E+P S D					
	E+	E+ S1 S2 D	• (S1+1, S1)+(S2+1, S2) → (D+1, D)		4	—	6-42
	E+P	E+P S1 S2 D					
	E-	E− S D	• (D+1, D)-(S+1, S) → (D+1, D)		3	—	6-40
	E-P	E−P S D					
	E-	E− S1 S2 D	• (S1+1, S1)-(S2+1, S2) → (D+1, D)		4	—	6-42
	E-P	E−P S1 S2 D					
Operaciones de multiplicación y división de datos decimales en coma flotante.	E*	E* S1 S2 D	• (S1+1, S1) * (S2+1, S2) → (D+1, D)		3	—	6-44
	E*P	E*P S1 S2 D					
	E/	E/ S1 S2 D	• (S1+1, S1)/(S2+1, S2) → Cociente (D+1, D)		4	—	6-44
	E/P	E/P S1 S2 D					
Operaciones de suma y resta de bloques binarios	BK+	BK+ S1 S2 D n	• Suma datos de n puntos desde (S1) y datos de n puntos hasta (S2) por lotes.		5	—	6-46
	BK+P	BK+P S1 S2 D n					
	BK-	BK− S1 S2 D n	• Resta datos de n puntos desde (S1) y datos de n puntos hasta (S2) por lotes.		5	—	6-46
	BK-P	BK−P S1 S2 D n					
Combinaciones de datos de cadenas de caracteres	$+	$+ S D	• Conecta la cadena de caracteres designada por (S) con la cadena de caracteres designada por (D), y almacena su resultado (D) posteriormente.		3	—	6-49
	$+P	$+P S D					
	$+	$+ S1 S2 D	• Conecta la cadena de caracteres designada por (S2) con la cadena de caracteres designada por (S1), y almacena su resultado (D) posteriormente.		4	—	6-51
	$+P	$+P S1 S2 D					

Tabla 2.11 Instrucciones de operaciones aritméticas (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Incremento datos BIN	INC	INC D	• (D)+1 → (D)		2	●	6-53
	INCP	INCP D					
	DINC	DINC D	• (D+1, D)+1 → (D+1, D)		*1	●	6-55
	DINCP	DINCP D					
	DEC	DEC D	• (D)-1 → (D)		2	●	6-53
	DECP	DECP D					
	DDEC	DDEC D	• (D+1, D)-1 → (D+1, D)		*1	●	6-55
	DDECP	DDECP D					

Comentarios

1) *1: El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU de procesos	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificación de índice. • Constante : Sin limitaciones	Nota 1) 3
	(2) En caso de utilización a parte de (1)	Nota 2) 2
Modelos QCPU básicos QnCPU	2	Nota 2)

Nota 1:Para el modelo QCPU o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

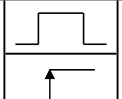
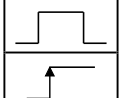
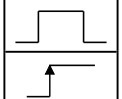
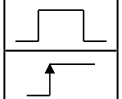
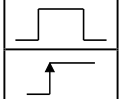
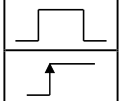
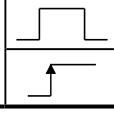
Nota 2:El número de pasos puede incrementar debido a las condiciones descritas en la sección 3.8.

2.4.3 Instrucciones de conversión de datos

Tabla 2.12 Instrucciones de conversión de datos

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Conversión BCD	BCD	BCD S D	Conversión BCD (S) → (D) BIN (0 a 9999)		3	●	6-57
	BCDP	BCDP S D					
	DBCD	DBCD S D	Conversión BCD (S+1, S) → (D+1, D) BIN (0 a 99999999)		3	●	6-57
	DBCDP	DBCDP S D					
Conversión binaria	BIN	BIN S D	Conversión BCD (S) → (D) BCD (0 a 9999)		3	●	6-59
	BINP	BINP S D					
	DBIN	DBIN S D	Conversión BCD (S+1, S) → (D+1, D) BCD (0 a 99999999)		3	●	6-59
	DBINP	DBINP S D					
Conversión de binario a decimal en coma flotante	FLT	FLT S D	Conversión a coma flotante (S+1, S) → (D) BIN (-32768 a 32767)		3	—	6-61
	FLTP	FLTP S D					
	DFLT	DFLT S D	Conversión a coma flotante (S+1, S) → (D+1, D) Número real (-2147483648 a 2147483647)		3	—	6-61
	DFLTP	DFLTP S D					
Conversión de decimal en coma flotante a binario	INT	INT S D	Conversión a Binario (S+1, S) → (D) Número real (-32768 a 32767)		3	—	6-63
	INTP	INTP S D					
	DINT	DINT S D	Conversión a Binario (S+1, S) → (D+1, D) Número real (-2147483648 a 2147483647)		3	—	6-63
	DINTP	DINTP S D					
Conversión entre binario de 16 bits y 32 bits	DBL	DBL S D	Conversión (S) → (D+1, D) BIN (-32768 a 32767)		3	—	6-65
	DBLP	DBLP S D					
	WORD	WORD S D	Conversión (S+1, S) → (D) BIN (-32768 a 32767)		3	—	6-66
	WORDP	WORDP S D					
Conversión de binario a gray	GRY	GRY S D	Conversión a código (S) → (D) BIN (-32768 a 32767)		3	—	6-67
	GRYP	GRYP S D					
	DGRY	DGRY S D	Conversión a código (S+1, S) → (D+1, D) BIN (-2147483648 a 2147483647)		3	—	6-67
	DGRYP	DGRYP S D					

Tabla 2.12 Instrucciones de conversión de datos (continuación)

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Conversión de código gray a binario	GBIN	GBIN S D	Conversión a dato BIN • (S) → (D) (S): Código Gray (-32768 a 32767)		3	—	6-69
	GBINP	GBINP S D					
	DGBIN	DGBIN S D	Conversión a dato BIN • (S+1, S) → (D+1, D) (S+1, S): Código Gray (-2147483648 a 2147483647)		3	—	6-69
	DGBINP	DGBINP S D					
Comple_ mento a2	NEG	NEG D	• $\overline{(D)}$ → (D) $\overline{(D)}$: Dato binario		2	—	6-71
	NEGP	NEGP D					
	DNEG	DNEG D	• $\overline{(D+1, D)}$ → (D+1, D) $\overline{(D+1, D)}$: Dato binario		2	—	6-71
	DNEGP	DNEGP D					
	ENEG	ENEG D	• $\overline{(D+1, D)}$ → (D+1, D) $\overline{(D+1, D)}$: Dato en número real		2	—	6-73
	ENEGP	ENEGP D					
Conversión en bloques	BKBCD	BKBCD S D n	• La cifra convierte n puntos de datos en BIN a datos BCD y almacena su resultado en (D) y posteriores.		4	—	6-74
	BKBCDP	BKBCDP S D n					
	BKBIN	BKBIN S D n	• Convierte n puntos de datos BCD de (S) a BIN y almacena el resultado en (D) y posteriores.		4	—	6-76
	BKBINP	BKBINP S D n					

2 - 23

2.4.4 Instrucciones de transferencia de datos.

Tabla 2.13 Instrucciones de transferencia de datos

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Transferencia de datos binarios de 16 bits	MOV	MOV S D	• (S) → (D)		*4	●	6-78
	MOVP	MOVP S D			*1		
Transferencia de datos binarios de 32 bits	DMOV	DMOV S D	• (S+1, S) → (D+1, D)		*2	●	6-78
	DMOVP	DMOVP S D					
Transferencia de datos decimales en coma flotante	EMOV	EMOV S D	• (S+1, S) → (D+1, D) Dato número real		*2	● *3	6-80
	EMOVP	EMOVP S D					
Transferencia de datos de cadenas de caracteres	$MOV	$MOV S D	• Transferencia de cadenas de caracteres designadas por (S) a dispositivo designado por (D) y posteriores.		3	—	6-82
	$MOVP	$MOVP S D					
Transferencia de datos negados de 16 bits	CML	CML S D	• $\overline{(S)}$ → (D)		*1	●	6-84
	CMLP	CMLP S D					
Transferencia de datos negados de 32 bits	DCML	DCML S D	• $\overline{(S+1, S)}$ → (D+1, D)		*2	●	6-84
	DCMLP	DCMLP S D					
Transferencia de bloques	BMOV	BMOV S D n	(S) → (D), n		4	●	6-87
	BMOVP	BMOVP S D n					
Transferencia múltiple de bloques de datos iguales	FMOV	FMOV S D n	(S) → (D), n		4	●	6-89
	FMOVP	FMOVP S D n					
Intercambio de datos de 16 bits	XCH	XCH S D	• (S) ↔ (D)		3	●	6-91
	XCHP	XCHP S D					
Intercambio de datos de 32 bits	DXCH	DXCH S D	• (S+1, S) ↔ (D+1, D)		3	●	6-91
	DXCHP	DXCHP S D					
Intercambio de bloques de datos	BXCH	BXCH S D n	(S) ↔ (D), n		4	—	6-93
	BXCHP	BXCHP S D n					
Intercambio entre bits altos y bits bajos	SWAP	SWAP D	(S) b15 to b8 / b7 to b0 (8 bits / 8 bits) (D) b15 to b8 / b7 to b0 (8 bits / 8 bits)		3	—	6-95
	SWAPP	SWAPP D					

Comentarios

1) ∗1: El Número de pasos puede variar según el dispositivo y tipo de modulo de CPU que sé este utilizando.

Componente	Número de pasos básicos	
QCPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de índice. • Constante : Sin limitaciones	2
	(2) En caso de utilización a parte de (1)	Nota (1) 3
QnACPU	3	Nota (1)

Nota 1: El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

2) ∗2: El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos básicos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de índice. • Constante : Sin limitaciones	2
	(2) En caso de utilización a parte de (1)	Nota (1) 3
Modelos Básicos QCPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de índice. • Constante : Sin limitaciones (El número de pasos es 3 cuando se utilizan los dispositivos y constantes anteriores).	2
	(2) En caso de utilización a parte de (1)	Nota (1) 3
QnCPU	3	Nota (1)

Nota 1: El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

3) ∗3 : El subprograma solo es efectivo con el modelo QCPU.

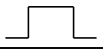
4) ∗4 : El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos básicos	
QCPU QnACPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de índice. • Constante : Sin limitaciones	3
	(2) En caso de utilización a parte de (1)	Nota (1) 3

Nota 1: El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

2.4.5 Instrucciones de ramificación de programa

Tabla 2.14 Instrucciones de ramificación de programa

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Salto	CJ	CJ Pn	• Salta a Pn cuando se cumplen las condiciones de entrada.		2	●	6-96
	SCJ	SCJ Pn	• Salta a Pn cuando se cumplen las condiciones de entrada en el ciclo de scan siguiente		2	●	6-96
	JMP	JMP Pn	• Salto incondicional a Pn		2	●	6-96
	GOEND	GOEND	• Salta a la instrucción END cuando se cumplen las condiciones de entrada.		1	—	6-99

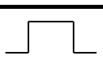
2.4.6 Instrucciones de control de ejecución de programa.

Tabla 2.15 Instrucciones de control de ejecución de programa

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Deshabilitación de interrupciones	DI	DI	• Prohíbe las interrupciones del programa.		1	—	6-100
Habilitación de interrupciones	EI	EI	• Permite la interrupción de programa.		1	—	6-100
Ajusta la Habilitación / Deshabilitación de interrupciones	IMASK	IMASK S	• Prohíbe o permite la interrupción para cada interrupción de programa.		2	—	6-100
Retorno	IRET	IRET	• Retorna a la secuencia de programa después de una interrupción de programa.		1	—	6-109

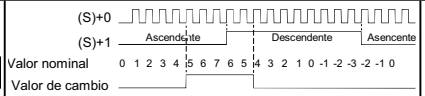
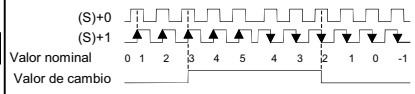
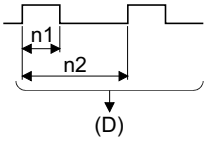
2.4.7 Instrucciones de refresco de E/S.

Tabla 2.16 Instrucciones de refresco de E/S

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Refresco E/S	RFS	RFS D n	• Refresco del área relevante de E/S durante el ciclo de scan.		3	—	6-111
	RFSP	RFSP D n					

2.4.8 Otras instrucciones relevantes

Tabla 2.17 Otras instrucciones relevantes.

Categoría	Nomenclatura	Símbolo	Detalles de procesamiento	Condición de ejecución	Número de pasos	Subprograma	Ver la descripción
Contador ascendente / descendente	UDCNT1	UDCNT1 S D n			4	—	6-113
	UDCNT2	UDCNT2 S D n			4	—	6-115
Temporizador de retentivo	TTMR	TTMR D n	• (Tiempo en que TTMR está a ON) * n → (D) n = 0:1, n = 1:10, n = 2:100		3	—	6-117
Temporizador especial	STMR	STMR S n D	• Los 4 puntos del bit de dispositivo designado por (D) funcionan tal y como se observa, dependiendo del estado de ON/OFF de las condiciones de entrada del STMR. instrucción: (D)+0: Activación a la desconexión del temporizador. (D)+1: Activación instantánea y desactivación al final de la temporización. (D)+2: desactivación instantánea y activación al final de la temporización. (D)+3: Activación a la conexión del temporizador.		3	—	6-119
Control por menor desplazamiento	ROTC	ROTC S n1 n2 D	• Rota una tabla rotatoria con n1 divisiones desde la señal de stop hasta la posición designada (S+1) por la posición más cercana.		5	—	6-122
Señal de rampa	RAMP	RAMP n1 n2 D1 n3 D2	• Cambia el dispositivo de salida designado por D1 desde n1 hasta n2 en n3 ciclos de scan.		6	—	6-124
Densidad de pulso	SPD	SPD S n D	• Cuenta la entrada de pulsos desde el dispositivo designada por (S) para la duración del tiempo designado por n y almacena el contaje en el dispositivo designado por (D).		4	—	6-126
Salida de pulsos	PLSY	PLSY n1 n2 D	• (n1)Hz → (D) Tiempo de salida n2		4	—	6-128
Modulación por anchura de pulsos	PWM	PWM n1 n2 D			4	—	6-130
Matriz de entrada	MTR	MTR S D1 D2 n	• Almacena 16 bits veces en n columnas en el dispositivo especificado por (S).al dispositivo especificado por (D2) correlativamente.		5	—	6-132

2.5 Instrucciones de Aplicación

2.5.1 Instrucciones de operaciones lógicas

Tabla 2.18 Instrucciones de operaciones lógicas

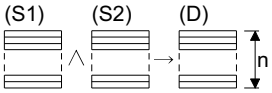
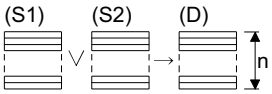
Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Producto lógico	WAND	WAND S D	• (D)∧(S)→(D)		3	●	7-3
	WANDP	WANDP S D					
	WAND	WAND S1 S2 D	• (S1)∧(S2)→(D)		4	● ＊3	7-5
	WANDP	WANDP S1 S2 D					
	DAND	DAND S D	• (D+1, D)∧(S+1, S)→(D+1, D)		＊1	●	7-3
	DANDP	DANDP S D					
	DAND	DAND S1 S2 D	• (S1+1, S1)∧(S2+1, S2)→(D+1, D)		＊2	● ＊3	7-5
	DANDP	DANDP S1 S2 D					
	BKAND	BKAND S1 S2 D n			5	—	7-8
	BKANDP	BKANDP S1 S2 D n					
Suma lógica	WOR	WOR S D	• (D)∨(S)→(D)		3	●	7-10
	WORP	WORP S D					
	WOR	WOR S1 S2 D	• (S1)∨(S2)→(D)		4	● ＊3	7-12
	WORP	WORP S1 S2 D					
	DOR	DOR S D	• (D+1, D)∨(S+1, S)→(D+1, D)		＊1	●	7-10
	DORP	DORP S D					
	DOR	DOR S1 S2 D	• (S1+1, S1)∨(S2+1, S2)→(D+1, D)		＊2	● ＊3	7-12
	DORP	DORP S1 S2 D					
	BKOR	BKOR S1 S2 D n			5	—	7-14
	BKORP	BKORP S1 S2 D n					
OR excluisva	WXOR	WXOR S D	• (D)⩛(S)→(D)		3	●	7-16
	WXORP	WXORP S D					
	WXOR	WXOR S1 S2 D	• (S1)⩛(S2)→(D)		4	● ＊3	7-18
	WXORP	WXORP S1 S2 D					

Tabla 2.18 Instrucciones de Operaciones Lógicas(Continuación)

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
OR-Exclusiva	DXOR	DXOR S D	• (D+1, D) ⩒ (S+1, S) → (D+1, D)		*1	●	7-16
	DXORP	DXORP S D					
	DXOR	DXOR S1 S2 D	• (S1+1, S1) ⩒ (S2+1, S2) → (D+1, D)		*2	● *3	7-18
	DXORP	DXORP S1 S2 D					
	BKXOR	BKXORP S1 S2 D n	(S1) (S2) (D) n		5	—	7-20
	BKXORP	BKXORP S1 S2 D n					
Suma lógica NO-OR exclusiva	WXNR	WXNR S D	• $\overline{(D) \veebar (S)} \rightarrow (D)$		3	●	7-22
	WXNRP	WXNRP S D					
	WXNR	WXNR S1 S2 D	• $\overline{(S1) \veebar (S2)} \rightarrow (D)$		4	● *3	7-25
	WXNRP	WXNRP S1 S2 D					
	DXNR	DXNR S D	• $\overline{(D+1, D) \veebar (S+1, S)} \rightarrow (D+1, D)$		*1	●	7-22
	DXNRP	DXNRP S D					
	DXNR	DXNR S1 S2 D	• $\overline{(S1+1, S1) \veebar (S2+1, S2)} \rightarrow (D+1, D)$		*2	● *3	7-25
	DXNRP	DXNRP S1 S2 D					
	BKXNR	BKXNR S1 S2 D n	$\overline{(S1)}$ $\overline{(S2)}$ (D) n		5	—	7-28
	BKXNRP	BKXNRP S1 S2 D n					

OBSERVACIÓN

1) ∗1:El Número de pasos puede variar según el dispositivo y tipo de modulo de CPU que se este utilizando.

Componente	Número de pasos básicos	
Modelo QCPU de Altas Prestaciones, CPU de proceso	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de inidice. • Constante : Sin limitaciones	Nota (1) 5
	(2) En caso de utilización a parte de (1)	Nota (2) 3
Modelos QCPU Básicos QnCPU	3	Nota (2)

Note 1:Para el modelo QCPU de alto nivel o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

Note 2:El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

2) ∗2: El número de pasos puede variar dependiendo del dispositivo y el tipo de CPU que se utilice.

Componente	Número de pasos básicos	
Modelo QCPU de Altas Prestaciones, Procesamiento CPU	(1) En caso de utilización de los siguientes dispositivos • Dispositivo Word : Dispositivos Internos (excepto registro ZR) • Dispositivo Bit : Dispositivos cuyo nº de dispositivo es múltiplo de 16, cuyo digito de designación es K8, y no utiliza modificador de indice. • Constante : Sin limitaciones	Nota (1) 6
	(2) En caso de utilización a parte de (1)	Nota (2) 4
Modelos Básicos QCPU QnCPU	4	Nota (2)

Note 1:Para el modelo QCPU de alto nivel o la CPU de proceso, el número de pasos es Mayor pero la velocidad de procesamiento es mayor en (1).

Note 2:El número de pasos puede incrementarse debido a las condiciones descritas en el apartado 3.8.

3) ∗3 : El subprograma es únicamente efectivo con el modelo QCPU.

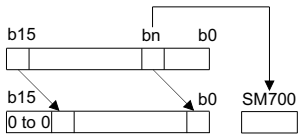
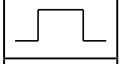
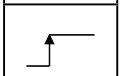
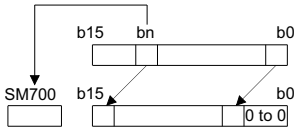
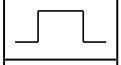
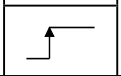
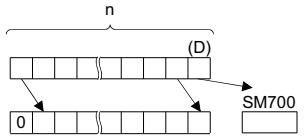
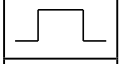
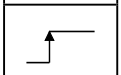
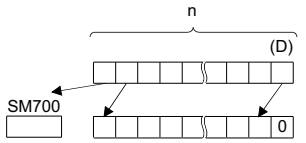
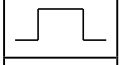
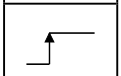
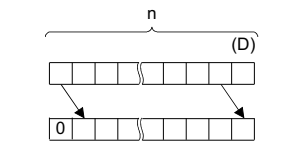
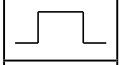
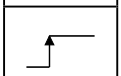
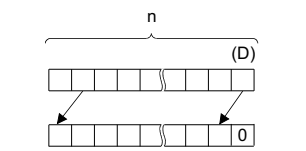
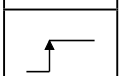
2.5.2 Instrucciones de rotación

Tabla 2.19 Instrucciones de rotación

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Rotación a la izquierda	ROR	ROR D n	b15 (D) b0 SM700 Rota n bits a la izquierda		3	●	7-30
	RORP	RORP D n					
	RCR	RCR D n	b15 (D) b0 SM700 Rota n bits a la izquierda con bit de acarreo		3	●	7-30
	RCRP	RCRP D n					
Rotación a la derecha	ROL	ROL D n	SM700 b15 (D) b0 Rota n bits a la derecha		3	●	7-32
	ROLP	ROLP D n					
	RCL	RCL D n	SM700 b15 (D) b0 Rota n bits a la derecha con bit de acarreo		3	●	7-32
	RCLP	RCLP D n					
Rotación a la izquierda	DROR	DROR D n	(D+1) (D) b31 to b16 b15 to b0 SM700 Rota n bits a la izquierda		3	●	7-34
	DRORP	DRORP D n					
	DRCR	DRCR D n	(D+1) (D) b31 to b16 b15 to b0 SM700 Rota n bits a la derecha con bit de acarreo		3	●	7-34
	DRCRP	DRCRP D n					
Rotación a la derecha	DROL	DROL D n	(D+1) (D) SM700 b31 to b16 b15 to b0 Rota n bits a la derecha		3	●	7-36
	DROLP	DROLP D n					
	DRCL	DRCL D n	(D+1) (D) SM700 b31 to b16 b15 to b0 Rota n bits a la derecha con bit de acarreo		3	●	7-36
	DRCLP	DRCLP D n					

2.5.3 Instrucciones de desplazamiento

Tabla 2.20 Instrucciones de desplazamiento

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Desplazamiento de n bits	SFR	SFR D n			3	●	7-38
	SFRP	SFRP D n					
	SFL	SFL D n			3	●	7-38
	SFLP	SFLP D n					
Desplazamiento de 1 bit	BSFR	BSFR D n			3	—	7-40
	BSFRP	BSFRP D n					
	BSFL	BSFL D n			3	—	7-40
	BSFLP	BSFLP D n					
Desplazamiento de 1 palabra	DSFR	DSFR D n			3	●	7-42
	DSFRP	DSFRP D n					
	DSFL	DSFL D n			3	●	7-42
	DSFLP	DSFLP D n					

2.5.4 Instrucciones de procesamiento de bits

Tabla 2.21 Instrucciones de procesamiento de bits

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Bit set / reset	BSET	BSET D n	(D) b15 bn b0 → 1		3	●	7-44
	BSETP	BSETP D n					
	BRST	BRST D n	(D) b15 bn b0 → 0		3	●	7-44
	BRSTP	BRSTP D n					
Bit de test	TEST	TEST S1 S2 D	(S1) b15 to b0 (D); Bit designado por (S2)		4	–	7-46
	TESTP	TESTP S1 S2 D					
	DTEST	DTEST S1 S2 D	(S1) b31 to b0 (D); Bit designado por (S2)		4	–	7-46
	DTESTP	DTESTP S1 S2 D					
Reset de dispositivos de bit por lotes	BKRST	BKRST S n	(S) ON/OFF ... ON/ON → Reset → (S) OFF/OFF ... OFF/OFF; n		3	–	7-48
	BKRSTP	BKRSTP S n					

2 - 33

2.5.5 Instrucciones de procesamiento de datos

Tabla 2.22 Instrucciones de procesamiento de datos

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Búsqueda de datos	SER	SER S1 S2 D n	(S1), (S2), n; (D) :Similitud No.; (D+1) :Número de similitudes		5	—	7-50
	SERP	SERP S1 S2 D n					
	DSER	DSER S1 S2 D n	32 bits; (S1), (S2), n; (D) :Similitud No.; (D+1) :Número de similitudes		5	—	7-50
	DSERP	DSERP S1 S2 D n					
Chequeo de bits	SUM	SUM S D	(S) b15 b0; (D): Numero de 1s		3	●	7-54
	SUMP	SUMP S D					
	DSUM	DSUM S D	(S+1) (S); (D): Numero de 1s		3	●	7-54
	DSUMP	DSUMP S D					
Decodificador	DECO	DECO S D n	Decodificador de 8 a 256; (S) Decodificador (D); n; 2^nbits		4	—	7-56
	DECOP	DECOP S D n					
Codificador	ENCO	ENCO S D n	Decodificador de 256 a 8; (S); 2^nbits; Codificador (D); n		4	—	7-58
	ENCOP	ENCOP S D n					
Decodificador 7 segmentos	SEG	SEG S D	b3 a b0; (S) (D); 7SEG		3	●	7-60
	SEGP	SEGP S D					
Separación y unión	DIS	DIS S D n	• Separa 16 bits designados por (S) en unidades de 4 bits y almacena a partir de los cuatro bits bajos de n puntos desde el la posición (D). (n ≤ 4)		4	—	7-62
	DISP	DISP S D n					
	UNI	UNI S D n	• Une los 4 bits bajos de n puntos desde el dispositivo designado por (S) y almacena en el dispositivo designado por (D). (n ≤ 4)		4	—	7-64
	UNIP	UNIP S D n					
	NDIS	NDIS S1 D S2	• Separa el dato de los dispositivos posteriores a (S1) en bits posteriores a (S2) y almacena secuencialmente a partir del dispositivo designado por (D).		4	—	7-66
	NDISP	NDISP S1 D S2					
	NUNI	NUNI S1 D S2	• Une el dato de los dispositivos posteriores a (S1) en bits posteriores a (S2) y almacena secuencialmente a partir del dispositivo designado por (D).				
	NUNIP	NUNIP S1 D S2					

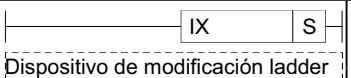
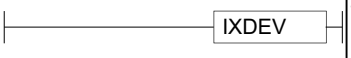
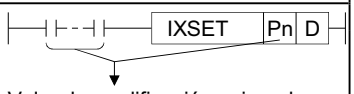
Tabla 2.22 Instrucciones de procesamiento de datos (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Separación y unión	WTOB	WTOB S D n	• Separa n-puntos de 16 bits de datos desde el dispositivo designado por (S) en 8 unidades y almacena secuencialmente en el dispositivo designado por (D).		4	—	7-71
	WTOBP	WTOBP S D n					
	BTOW	BTOW S D n	• Une los 8 bits bajos de 16 bits de datos de n-puntos desde el dispositivo designado por (S) en 16 unidades de bit y almacena secuencialmente en el dispositivo designado por (D).				
	BTOWP	BTOWP S D n					
Búsqueda	MAX	MAX S D n	• Busca el dato de n-puntos desde el dispositivo designado por (S) en unidades de 16 bits y almacena el valor máximo en el dispositivo designado por (D).		4	—	7-75
	MAXP	MAXP S D n					
	MIN	MIN S D n	• Busca el dato de n-puntos desde el dispositivo designado por (S) en unidades de 16 bits y almacena el valor mínimo en el dispositivo designado por (D).			—	7-77
	MINP	MINP S D n					
	DMAX	DMAX S D n	• Busca el dato de 2 * n-puntos desde el dispositivo designado por (S) en unidades de 32 bits y almacena el máximo valor del dispositivo designado por (D).		4	—	7-75
	DMAXP	DMAXP S D n					
	DMIN	DMIN S D n	• Busca el dato de 2 * n-puntos desde el dispositivo designado por (S) en unidades de 32 bits y almacena el mínimo valor del dispositivo designado por (D).				7-77
	DMINP	DMINP S D n					
Ordenación	SORT	SORT S1 n S2 D1 D2 • S2: Número de comparaciones hechas durante un ciclo • D1: Dispositivo de commutación cuando la clasificación es completada • D2: Para el uso del sistema	• Clasifica datos de n-puntos desde dispositivos designados por (S1) en unidades de 16 bits. (n x (n-1)/2 ciclos de scan requeridos)		6	—	7-80
	DSORT	DSORT S1 n S2 D1 D2 • S2: Número de comparaciones hechas durante un ciclo • D1: Dispositivo de commutación cuando la clasificación es completada • D2: Para el uso del sistema	• Clasifica datos de 2 * n puntos desde el dispositivo designado por (S1) en unidades de 32 bits. (n x (n+1)/2 ciclos de scan requeridos)				
Cálculos totales evaluados	WSUM	WSUM S D n	• Suma 16 bits en datos binarios de n puntos desde el dispositivo designado por (S) y lo almacena en el dispositivo especificado por (D).		4	—	7-83
	WSUMP	WSUMP S D n					
	DWSUM	DWSUM S D n	• Suma 32 bits en datos binarios de n puntos desde el dispositivo designado por (S) y lo almacena en el dispositivo especificado por (D).				7-85
	DWSUMP	DWSUMP S D n					

2.5.6 Instrucciones de creación de estructuras

Tabla 2.23 Instrucciones de creación de estructuras

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Número de repeticiones	FOR	FOR n	• Ejecuta n veces entre FOR y NEXT		2	—	7-87
	NEXT	NEXT			1	—	
	BREAK	BREAK D Pn	• Fuerza la salida de ejecución del FOR hasta el ciclo NEXT y salta al puntero Pn.		3	—	7-89
	BREAKP	BREAKP D Pn					
Llamadas a subrutinas de programa	CALL	CALL Pn S1 to Sn	• Ejecuta la subrutina de programa Pn cuando se cumple la condición de entrada. (De S1 a Sn son argumentos de envío de subrutina de programa. 0 ≤ n ≤ 5)		*1 2 + n	—	7-91
	CALLP	CALLP Pn S1 to Sn					
	RET	RET	• Retorno a la subrutina de programa		1	—	7-95
	FCALL	FCALL Pn S1 to Sn	• Realiza el procesamiento de no operación de la subrutina de programa Pn si la condición de entrada no se cumple. (De S1 a Sn son argumentos de envío de subrutina de programa. 0 ≤ n ≤ 5)		*1 2 + n	—	7-96
	FCALLP	FCALLP Pn S1 to Sn					
	ECALL	ECALL * Pn S1 to Sn *: Nombre del programa	• Ejecuta la subrutina de programa Pn desde dentro del programa designado cuando la condición de entrada se cumple. (De S1 a Sn son argumentos de envío de subrutina de programa. 0 ≤ n ≤ 5)		*2 3 + n	—	7-100
	ECALLP	ECALLP * Pn S1 to Sn *: Nombre del programa					
	EFCALL	EFCALL * Pn S1 to Sn *: Nombre del programa	• Realiza el procesamiento de no operación de la subrutina de programa Pn desde dentro del programa designado si la condición de entrada no se cumple. (De S1 a Sn son argumentos de envío de subrutina de programa. 0 ≤ n ≤ 5)		*2 3 + n	—	7-104
	EFCALLP	EFCALLP * Pn S1 to Sn *: Nombre del programa					
	XCALL	XCALL Pn S1 to Sn	• Ejecuta la subrutina de programa Pn cuando la condición de entrada no se cumple. • Realiza la no ejecución de procesamiento de la subrutina de programa Pn cuando la condición de entrada no se cumple. (De S1 a Sn son argumentos de envío de subrutina de programa. 0 ≤ n ≤ 5)		*1 2 + n	—	7-108
	COM	COM	• Realiza el refresco de comunicaciones y el procesamiento general de datos.		1	—	7-112

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Modificación de índice fijado	IX		• Dirige el índice de modificación para dispositivos individuales usados en dispositivos de modificación ladder.		2	—	7-118
	IXEND				1	—	
	IXDEV		• Almacena la modificación del valor usado para el índice de modificación realizado entre IX y IXEND en el dispositivo posterior designado por (D).		1	—	7-126
	IXSET				3	—	

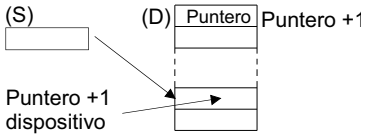
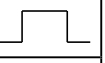
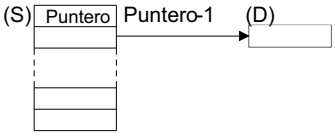
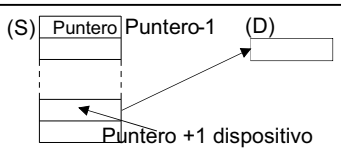
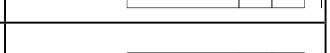
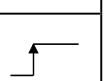
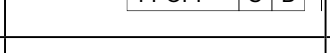
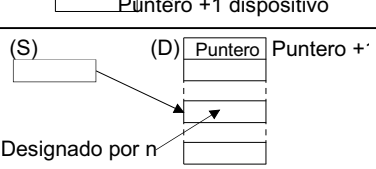
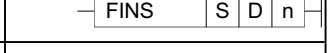
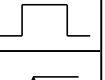
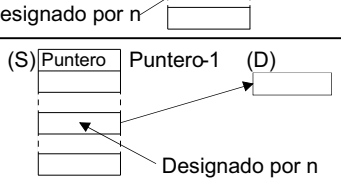
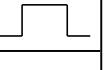
＊1 : n indica en número de argumentos por subrutina de programa.

＊2 : n indica el total de número de argumentos usados en la subrutina de programa y el número de pasos de nombre de programa.

El número de pasos de nombre de programa es calculado como "numero de caracteres en el programa/ 2" (las fracciones decimales son redondeadas).

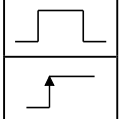
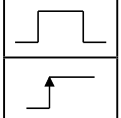
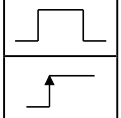
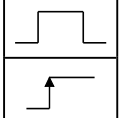
2.5.7 Instrucciones de operación de tabla de datos

Tabla 2.24 Instrucciones de operación de tabla de datos

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Procesamiento de tabla de datos	FIFW				3	—	7-131
	FIFWP						
	FIFR				3	—	7-133
	FIFRP						
	FPOP				3	—	7-135
	FPOPP						
	FINS				4	—	7-137
	FINSP						
	FDEL				4	—	7-137
	FDELP						

2.5.8 Instrucciones de acceso a buffer de memoria

Tabla 2.25 Instrucciones de acceso a buffer de memoria

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver la Descripción
Lectura de datos	FROM	FROM n1 n2 D n3	• Lee datos en unidades de 16 bits desde módulos de función inteligente / módulos de función especial.		5	—	7-140
	FROMP	FROMP n1 n2 D n3					
	DFRO	DFRO n1 n2 D n3	• Lee datos en unidades de 32 bits desde módulos de función inteligente / módulos de función especial.		5	—	7-140
	DFROP	DFROP n1 n2 D n3					
Escritura de datos	TO	TO n1 n2 S n3	• Escribe datos de unidades de 16 bits a módulos de función inteligente / módulos de función especial.		5	—	7-143
	TOP	TOP n1 n2 S n3					
	DTO	DTO n1 n2 S n3	• Escribe datos de unidades de 32 bits a módulos de función inteligente / módulos de función especial.		5	—	7-143
	DTOP	DTOP n1 n2 S n3					

2.5.9 Instrucciones de Visualización

Tabla 2.26 Instrucciones de Visualización

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Buscar por Descripción
Escritura ASCII	PR	* SM701 cuando está en ON PR S D	• Salida de código de 8 puntos ASCII (16 caracteres) desde el dispositivo designado por (S) a la salida del módulo.		3	—	7-146
	PR	* SM701 cuando está en OFF PR S D	• Salida de código ASCII desde el dispositivo designado por (S) hasta el código nulo 00_H es enviada la salida del módulo.				
	PRC	PRC S D	• Convierte los comentarios desde el dispositivo designado por (S) a código ASCII y entrega el resultado a la salida del módulo.				7-149
Display	LED	LED S	• Muestra el código ASCII de 8 puntos (16 caracteres) desde el dispositivo designado por (S) al dispositivo display LED que se ubica en frente del módulo de la CPU .		2	—	7-154
	LEDC	LEDC S	• Muestra los comentarios desde el dispositivo designado por (S) al dispositivo display LED que se ubica en frente del módulo de la CPU .				7-156
Reset	LEDR	LEDR	• Resetea el anunciador y el display de la unidad del display.		1	—	7-158

2.5.10 Instrucciones de diagnostico de fallos y depuración

Tabla 2.27 Instrucciones de diagnostico de fallos y depuración

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Chequeo	CHKST	CHKST	• La instrucción CHK es ejecutada cuando CHKST es ejecutado. • Salta al comando seguido de la instrucción CHK cuando CHKST se encuentra en el estado de no ejecución.		1	—	7-161
	CHK	CHK Check Condition	• En condiciones normales→ SM80: OFF, SD80: 0 • En condiciones de fallo → SM80: ON, SD80: Número de Fallos				
	CHKCIR	CHKCIR	• Empieza la actualización en modo ladder siendo chequeado por la instrucción CHK.		1	—	7-165
	CHKEND	CHKEND	• Acaba la actualización en modo ladder siendo chequeado por la instrucción CHK.				
Registro de estado retentivo	SLT	SLT	• Ejecuta el registro de estado retentivo		1	—	7-173
	SLTR	SLTR	• Resetea el registro de estado para habilitar la reejecución.				
Trazado de muestreo	STRA	STRA	• Aplica el disparo para trazar el programa.		1	—	7-175
	STRAR	STRAR	• Resetea la señal de muestreo para habilitar la reejecución.				
Trazado del Programa	PTRA	PTRA	• Aplica el disparo para trazado del programa		1	—	7-177
	PTRAR	PTRAR	• Resetea la señal del programa para habilitar la reejecución.				
	PTRAEXE	PTRAEXE	• Ejecuta trazado del programa.		1	—	7-177
	PTRAEXEP	PTRAEXEP					

2.5.11 Instrucciones de procesamiento de cadenas de caracteres

Tabla 2.28 Instrucciones de procesamiento de cadenas de caracteres

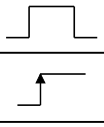
Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver Manual de programación
BIN a decimal de ASCII	BINDA	BINDA S D	• Conversión de 1 palabra en valor BIN designado por (S) a 5-dígitos, en decimal correspondiente del código ASCII y lo almacena en una palabra del dispositivo designada por (D).		3	—	7-179
	BINDAP	BINDAP S D					
	DBINDA	DBINDA S D	• Convierte 2 palabras de valor BIN designado por (S) a 10 dígitos de valor decimal correspondiente a código ASCII y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-179
	DBINDAP	DBINDAP S D					
BIN a hexa-decimal de ASCII	BINHA	BINHA S D	• Convierte 1 palabra de valor BIN designado por (S) a 4 dígitos de valor hexadecimal correspondiente a código ASCII y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-182
	BINHAP	BINHAP S D					
	DBINHA	DBINHA S D	• Convierte 2 palabras de valor BIN designado por (S) a 8 dígitos de valor hexadecimal correspondiente a código ASCII y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-182
	DBINHAP	DBINHAP S D					
BCD a decimal de ASCII	BCDDA	BCDDA S D	• Convierte 1 palabra en BCD designado por (S) a 4 dígitos de valor decimal correspondiente a código ASCII y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-185
	BCDDAP	BCDDAP S D					
	DBCDDA	DBCDDA S D	• Convierte 2 palabras en BCD designado por (S) a 8 dígitos de valor decimal correspondiente a código ASCII y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-185
	DBCDDAP	DBCDDAP S D					
Decimal de ASCII a BIN	DABIN	DABIN S D	• Convierte 5 dígitos de valor decimal correspondiente a código ASCII designado por (S) a 1 palabra de valor BIN y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-188
	DABINP	DABINP S D					
	DDABIN	DDABIN S D	• Convierte 10 dígitos de valor decimal en ASCII designado por (S) a 2 palabras de valor BIN y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-188
	DDABINP	DDABINP S D					
Hexadeci-mal de ASCII a BIN	HABIN	HABIN S D	• Convierte 4 dígitos de valor hexadecimal correspondiente a código ASCII designado por (S) a 1 palabra de valor BIN y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-191
	HABINP	HABINP S D					
	DHABIN	DHABIN S D	• Convierte 8 dígitos de valor BIN designado por (S) a 2 palabras de valor BIN y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-191
	DHABINP	DHABINP S D					

Tabla 2.28 Instrucciones de procesamiento de cadenas de caracteres (Continuación)

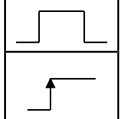
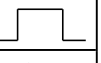
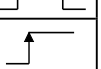
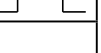
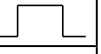
Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Decimal de ASCII a BCD	DABCD	DABCD S D	• Convierte 4 dígitos de valor decimal correspondiente a código ASCII designado por (S) a 1 palabra de valor BCD y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-193
	DABCDP	DABCDP S D					
	DDABCD	DDABCD S D	• Convierte 8 dígitos en decimal correspondiente a código ASCII designado por (S) a 2 palabras de valor BCD y lo almacena en una palabra de dispositivo designada por (D).		3	—	7-193
	DDABCDP	DDABCDP S D					
Operación de lectura comentario dispositivo	COMRD	COMRD S D	• Almacena el comentario del dispositivo designado por(S) al dispositivo designado por (D).		3	—	7-196
	COMRDP	COMRDP S D					
Detección de la longitud de cadena de caracteres	LEN	LEN S D	• Almacena la longitud del dato (número de caracteres) en la cadena de caracteres designada por (S) hasta el dispositivo designado por (D).		3	—	7-201
	LENP	LENP S D					
BIN a decimal de cadena de caracteres	STR	STR S1 S2 D	• Convierte 1 palabra de valor BIN designada por (S2) a decimal correspondiente a cadena de caracteres con un número total de dígitos y el número de dígitos de fracciones decimales designada por (S1) y lo almacena al dispositivo designado por (D).		4	—	7-203
	STRP	STRP S1 S2 D					
	DSTR	DSTR S1 S2 D	• Convierte 2 palabras de valor BIN designada por (S2) a decimal correspondiente a cadena de caracteres con un número total de dígitos y el número de dígitos de fracciones decimales designada por (S1) y lo almacena al dispositivo designado por (D).		4	—	7-203
	DSTRP	DSTRP S1 S2 D					
Cadenas de caracteres decimales a BIN	VAL	VAL S D1 D2	• Convierte una cadena de caracteres incluyendo el punto decimal designado por (S) a 1 palabra de valor BIN y el número de dígitos de fracción decimal y los almacena en dispositivos designados por (D1) y (D2).		4	—	7-209
	VALP	VALP S D1 D2					
	DVAL	DVAL S D1 D2	• Convierte una cadena de caracteres incluyendo el punto decimal designado por (S) a 2 palabras de valor BIN y el número de dígitos de fracción decimal y los almacena en dispositivos designados por (D1) y (D2).		4	—	7-209
	DVALP	DVALP S D1 D2					
Decimal en coma flotante a cadenas de caracteres	ESTR	ESTR S1 S2 D	• Convierte datos de decimal en coma flotante designados por (S1) a cadenas de caracteres y lo almacena en el dispositivo designado por (D).		4	—	7-214
	ESTRP	ESTRP S1 S2 D					
Cadena de caracteres a decimal en coma flotante	EVAL	EVAL S D	• Convierte cadenas de caracteres designadas por (S) a datos decimales en coma flotante y lo almacena en el dispositivo designado por (D).		3	—	7-221
	EVALP	EVALP S D					

Tabla 2.28 Instrucciones de procesamiento de cadenas de caracteres (Continuación)

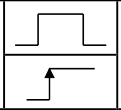
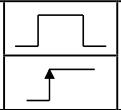
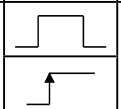
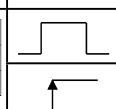
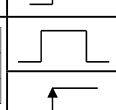
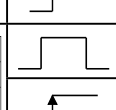
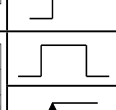
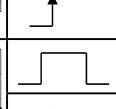
Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Hexadecimal BIN a ASCII	ASC	ASC S D n	• Convierte 1 palabra de valor BIN del número de dispositivo designado por (S) a ASCII, y almacena el resultado dentro del rango determinado por n a partir del dispositivo designado por (D).		4	—	7-225
	ASCP	ASCP S D n					
ASCII a hexadecimal BIN	HEX	HEX S D n	• Convierte n caracteres ASCII del número de dispositivo designado por (S) a valor BIN, y almacena el resultado el dispositivo designado por (D) dentro del rango determinado por n.		4	—	7-227
	HEXP	HEXP S D n					
Procesamiento de cadenas de caracteres	RIGHT	RIGHT S D n	• Almacena n caracteres desde el final de la cadena de caracteres designada por (S) hasta el dispositivo designado por (D).		4	—	7-229
	RIGHTP	RIGHTP S D n					
	LEFT	LEFT S D n	• Almacena n caracteres desde el principio de la cadena de caracteres designada por (S) hasta el dispositivo designado por (D).				
	LEFTP	LEFTP S D n					
	MIDR	MIDR S1 D S2	• Almacena el número designado de caracteres en la cadena de caracteres designada por (S1) desde la posición designada por (S2) al dispositivo indicado por (D).		4	—	7-232
	MIDRP	MIDRP S1 D S2					
	MIDW	MIDW S1 D S2	• Almacena el número designado de caracteres en la cadena de caracteres designada por (S1) desde la posición designada por (S2) al dispositivo indicado por (D).				
	MIDWP	MIDWP S1 D S2					
	INSTR	INSTR S1 S2 D n	• Busca la cadena de caracteres (S1) desde el carácter n de la cadena de caracteres (S2), y almacena el resultado en (D).		5	—	7-236
	INSTRP	INSTRP S1 S2 D n					
Decimal en coma flotante a BCD	EMOD	EMOD S1 S2 D	• Convierte datos decimales en coma flotante (S1) a datos en BCD con números de dígitos de fracciones decimales designado por (S2) , y almacena el resultado en le dispositivo designado por(D).		4	—	7-238
	EMODP	EMODP S1 S2 D					
BCD a decimal en coma flotante	EREXP	EREXP S1 S2 D	• Convierte a datos en BCD (S1) a datos decimales en coma flotante con números de dígitos de fracciones decimales designado por (S2), y almacena el resultado en le dispositivo designado (D).		4	—	7-240
	EREXPP	EREXPP S1 S2 D					

2.5.12 Instrucciones de función especiales

Tabla 2.29 Instrucción de funciones especiales

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Funciones Trígono_ métricas (Datos decimales en coma flotante)	SIN	SIN S D	• Sin (S+1, S) → (D+1, D)		3	—	7-242
	SINP	SINP S D					
	COS	COS S D	• Cos (S+1, S) → (D+1, D)		3	—	7-244
	COSP	COSP S D					
	TAN	TAN S D	• Tan (S+1, S) → (D+1, D)		3	—	7-246
	TANP	TANP S D					
	ASIN	ASIN S D	• Sin^{-1} (S+1, S) → (D+1, D)		3	—	7-248
	ASINP	ASINP S D					
	ACOS	ACOS S D	• Cos^{-1} (S+1, S) → (D+1, D)		3	—	7-250
	ACOSP	ACOSP S D					
	ATAN	ATAN S D	• Tan^{-1} (S+1, S) → (D+1, D)		3	—	7-252
	ATANP	ATANP S D					
Conversión entre ángulos y radianes	RAD	RAD S D	• (S+1, S) → (D+1, D) Conversión de Ángulos a radianes		3	—	7-254
	RADP	RADP S D					
	DEG	DEG S D	• (S+1, S) → (D+1, D) Conversión de Radianes a ángulos		3	—	7-256
	DEGP	DEGP S D					
Raíces cuadradas	SQR	SQR S D	• $\sqrt{(S+1, S)}$ → (D+1, D)		3	—	7-258
	SQRP	SQRP S D					
Operaciones exponenciales	EXP	EXP S D	• $e^{(S+1, S)}$ → (D+1, D)		3	—	7-260
	EXPP	EXPP S D					
Logaritmos naturales	LOG	LOG S D	• Log e (S+1, S) → (D+1, D)		3	—	7-262
	LOGP	LOGP S D					
Generación de números aleatoriamente	RND	RND D	• Genera un número aleatorio (de 0 a un número menor que 32767) y almacena el resultado en el dispositivo designado por (D).		2	—	7-264
	RNDP	RNDP D					
Actualización de series numéricas aleatorias	SRND	SRND S	• Actualiza las series numéricas aleatorias en concordancia con el dato de 16-bit BIN almacenado en el dispositivo designado por (S).				
	SRNDP	SRNDP S					

Tabla 2.29 Instrucciones de función especial (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Raíces cuadradas	BSQR	BSQR S D	• $\sqrt{(S)}$ → (D)+0 Parte entera; +1 Parte de la fracción decimal		3	—	7-266
	BSQRP	BSQRP S D					
	BDSQR	BDSQR S D	• $\sqrt{(S+1, S)}$ → (D)+0 Parte entera; +1 Parte fracción decimal		3	—	7-266
	BDSQRP	BDSQRP S D					
Funciones trígono_ métricas	BSIN	BSIN S D	• Sin (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-269
	BSINP	BSINP S D					
	BCOS	BCOS S D	• Cos (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-271
	BCOSP	BCOSP S D					
	BTAN	BTAN S D	• Tan (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-273
	BTANP	BTANP S D					
	BASIN	BASIN S D	• Sin^{-1} (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-275
	BASINP	BASINP S D					
	BACOS	BACOS S D	• Cos^{-1} (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-277
	BACOSP	BACOSP S D					
	BATAN	BATAN S D	• Tan^{-1} (S) → (D)+0 Signo; +1 Parte entera; +2 Parte de la fracción decimal		3	—	7-279
	BATANP	BATANP S D					

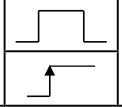
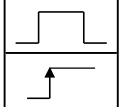
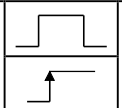
2.5.13 Instrucciones de control de datos

Tabla 2.30 Instrucciones de control de datos

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Control de límites superiores e inferiores	LIMIT	LIMIT S1 S2 S3 D	• Cuando (S3) < (S1) Almacena valor de (S1) en (D) • Cuando (S1) ≤ (S3) ≤ (S2) Almacena valor de (S3) en (D) • Cuando (S2) < (S3) Almacena valor de (S2) en (D)		5	—	7-281
	LIMITP	LIMITP S1 S2 S3 D					
	DLIMIT	DLIMIT S1 S2 S3 D	• Cuando ((S3)+1, (S3)) < ((S1)+1, S1) ....Almacena valor de ((S1)+1, (S1)) en ((D)+1, (D)) • Cuando ((S1)+1, (S1)) ≤ ((S3)+1, (S3))< (S2+1, S2) ....Almacena valor de ((S3)+1, (S3)) en ((D)+1, (D)) • Cuando ((S2), (S2)+1) < ((S3), (S3)+1) ....Almacena valor de ((S2)+1, (S2)) en ((D)+1, (D))		5	—	7-281
	DLIMITP	DLIMITP S1 S2 S3 D					
Control de zonas muertas	BAND	BAND S1 S2 S3 D	• Cuando (S1) ≤ (S3) ≤ (S2) 0 → (D) • Cuando (S3) < (S1)..... (S3)-(S1) → (D) • Cuando (S2) < (S3)..... (S3)-(S2) → (D)		5	—	7-284
	BANDP	BANDP S1 S2 S3 D					
	DBAND	DBAND S1 S2 S3 D	• Cuando ((S1)+1, (S1)) ≤ ((S3)+1, (S3)) ≤ ((S2)+1, (S2)) ....0 → ((D)+1, (D)) • Cuando ((S3)+1, (S3)) < ((S1)+1, (S1)) .. ((S3)+1, (S3)) - ((S1)+1, (S1)) → ((D)+1, (D)) • Cuando ((S2)+1, (S2)) < ((S3)+1, (S3)) .. ((S3)+1, (S3)) - ((S2)+1, (S2)) → ((D)+1, (D))		5	—	7-284
	DBANDP	DBANDP S1 S2 S3 D					
Control de zonas	ZONE	ZONE S1 S2 S3 D	• Cuando (S3) = 0.....0 → (D) • Cuando (S3) > 0.....(S3)+(S2) → (D) • Cuando (S3) < 0.....(S3)-(S1) → (D)		5	—	7-287
	ZONEP	ZONEP S1 S2 S3 D					
	DZONE	DZONE S1 S2 S3 D	• Cuando ((S3)+1, (S3)) = 0 ....0 → ((D)+1, (D)) • Cuando ((S3)+1, (S3)) > 0 ....((S3)+1, (S3))+((S2)+1, (S2)) → ((D)+1, (D)) • Cuando ((S3)+1, (S3)) < 0 ....((S3)+1, (S3)) + ((S1)+1, (S1)) → ((D)+1, (D))		5	—	7-287
	DZONEP	DZONEP S1 S2 S3 D					

2.5.14 Instrucciones de conmutado

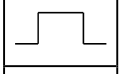
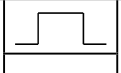
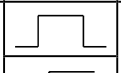
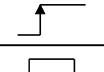
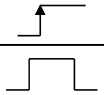
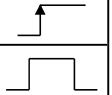
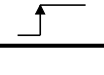
Tabla 2.31 Instrucciones de conmutado

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Designación de bloques numéricos	RSET	RSET S	• Conversión de bloques numéricos registros de archivos de extensión a números designados por (S).		2	—	7-290
	RSETP	RSETP S					
Habilitación de archivos	QDRSET	QDRSET File Name	• Habilitación nombres de archivos usados como registros de archivos.		∗2+n	—	7-292
	QDRSETP	QDRSETP File Name					
	QCDSET	QCDSET File Name	• Habilitación nombres de archivos usados como archivos.		∗2+n	—	7-294
	QCDSETP	QCDSETP File Name					

∗ : n ([número de archivos de caracteres] / 2) indica el paso de programa. (Las fracciones decimales son redondeadas en exceso.)

2.5.15 Instrucciones de Reloj(Reloj de Tiempo Real-RTC)

Tabla 2.32 Instrucciones de Reloj

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Lectura / Escritura de datos RTC	DATERD	DATERD D	• (Dsipositivo de reloj) → (D)+0 Año, +1 Mes, +2 Día, +3 Hora, +4 Minuto, +5 Seg, +6 Día de semana		2	—	7-296
	DATERDP	DATERDP D					
	DATEWR	DATEWR S	• (D)+0 Año, +1 Mes, +2 Dia, +3 Hora, +4 Minuto, +5 Seg., +6 Día de semana → (Dispositivo RTC)		2	—	7-300
	DATEWRP	DATEWRP S					
Suma / Resta de datos RTC	DATE+	DATE+ S1 S2 D	(S1) Hora, Minuto, Seg. + (S2) Hora, Minuto, Seg. → (D) Hora, Minuto, Seg.		4	—	7-304
	DATE+P	DATE+P S1 S2 D					
	DATE-	DATE− S1 S2 D	(S1) Hora, Minuto, Seg. - (S2) Hora, Minuto, Seg. → (D) Hora, Minuto, Seg.		4	—	7-306
	DATE-P	DATE−P S1 S2 D					
Traducción de datos RTC	SECOND	SECOND S D	(S) Hora, Minuto, Seg. → (D) Seg. (nivel inferior), Seg. (nivel superior)		3	—	7-308
	SECONDP	SECONDP S D					
	HOUR	HOUR S D	(S) Seg. (nivel inferior), Seg. (nivel superior) → (D) Hora, Minuto, Seg.				
	HOURP	HOURP S D					

2.5.16 Instrucciones de dispositivos periféricos

Tabla 2.33 Instrucciones de dispositivos de periféricos

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de Pasos	Subprograma	Ver manual de programación
Entrada / Salida dispositivos periféricos	MSG	MSG S	• Almacena el mensaje designado por (S) en QnACPU. Este mensaje es visualizado en el dispositivo periférico.		2	—	7-310
	PKEY	PKEY D	• Los datos de entrada desde el dispositivo periférico son almacenados en el dispositivo designado por (D).		2	—	7-312

2.5.17 Instrucciones de control de programa

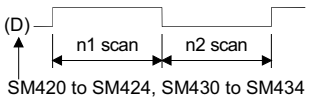
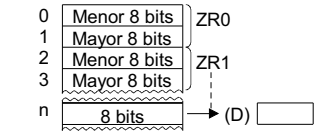
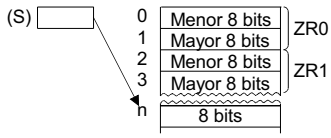
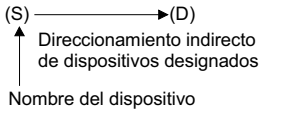
Tabla 2.34 Instrucciones de control de programa

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Instrucciones de control de programa	PSTOP	PSTOP Nombre programa	• Emplaza el programa designado en estado de mantenimiento		∗2+n	—	7-315
	PSTOPP	PSTOPP Nombre programa					
	POFF	POFF Nombre programa	• Apaga el contacto del programa designado en estado off y pone el programa en estado de mantenimiento.		∗2+n	—	7-316
	POFFP	POFFP Nombre programa					
	PSCAN	PSCAN Nombre programa	• Ejecuta el programa en modo scan.		∗2+n	—	7-318
	PSCANP	PSCANP Nombre programa					
	PLOW	PLOW Nombre programa	• Ejecuta el programa en modo baja velocidad.		∗2+n	—	7-320
	PLOWP	PLOWP Nombre programa					
	LDPCHK	PCHK Nombre programa	• Conmuta cuando el fichero del programa especificado está siendo ejecutado. • No conmuta cuando el programa del fichero especificado no está siendo ejecutado.		∗2+n	—	7-310
	ANDPCHK	PCHK Nombre programa					
	ORPCHK	PCHK Nombre programa					

∗ : n ([Número de caracteres de programa] / 2) indica los pasos de programa. (Las fracciones decimales son redondeadas al alza.)

2.5.18 Otras instrucciones

Tabla 2.35 Otras instrucciones

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
WDT reset	WDT	WDT	• Resetea el temporizador perro guardián durante la secuencia de programa		1	—	7-324
	WDTP	WDTP					
Duty cicle	DUTY	DUTY n1 n2 D			4	—	7-326
Tempori_zador a la conexión	TIMCHK	TIMCHK S1 S2 D	• Conmuta en dispositivo designado por (D) si se la temporización excede del tiempo designado.			—	7-328
Lectura / Escritura de operaciones de unidades de 1 bit	ZRRDB	ZRRDB n D			3	—	7-329
	ZRRDBP	ZRRDBP n D					
	ZRWRB	ZRWRB n S			3	—	7-331
	ZRWRBP	ZRWRBP n S					
	ADRSET	ADRSET S D			3	—	7-333
	ADRSETP	ADRSETP S D					
Clave de entrada numérica para teclado	KEY	KEY S n D1 D2	• Adquiere datos en ASCII para 8 puntos de unidades de entrada designados por (S), convierte a hexadecimal valores siguiendo el número de dispositivo designado por (D1), y lo almacena.		5	—	7-334
Almacenamiento por lotes de índices de registro	ZPUSH	ZPUSH D	• Salva el contenido de los índices de registro Z0 hasta Z15 a partir del dispositivo designado por (D).		2	—	7-338
	ZPUSHP	ZPUSHP D					
Carga de lotes de índices de registro	ZPOP	ZPOP D	• Carga el contenido de los índices de registro Z0 hasta Z15 a partir del dispositivo designado por (D).				
	ZPOPP	ZPOPP D					
Escritura de lotes de operación a registros de archivo E^2PROM	EROMWR	EROMWR S D1 n D2	• Escribe al lote de datos de registros de archivo E^2PROM.		5	—	7-342
	EROMWRP	EROMWRP S D1 n D2					

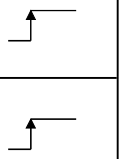
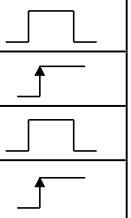
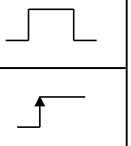
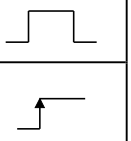
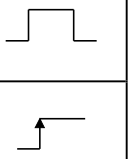
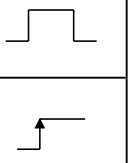
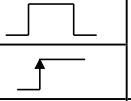
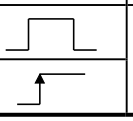
2 - 49

2.5.19 Instrucciones para Comunicación de Datos

Tabla 2.36 Instrucciones para Comunicación de Datos

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Refresco de red	ZCOM	J.ZCOM Jn JP.ZCOM Jn G.ZCOM Un GP.ZCOM Un	Refresca la red designada		5	—	8-76
Instrucciones de QnA link Lectura de datos desde otra estación	READ	J.READ Jn S1 S2 D1 D2 G.READ Jn S1 S2 D1 D2 JP.READ Jn S1 S2 D1 D2 GP.READ Un S1 S2 D1 D2	Lee la palabra de datos del dispositivo de otra estación para otra estación anfitrión (host station)		9	—	8-15
	SREAD	J.SREAD Jn S1 S2 D1 D2 D3 G.SREAD Un S1 S2 D1 D2 D3 JP.SREAD Jn S1 S2 D1 D2 D3 GP.SREAD Un S1 S2 D1 D2 D3			10	—	8-21
Instrucciones de QnA link Escritura de datos desde otra estación	WRITE	J.WRITE Jn S1 S2 D1 D2 G.WRITE Un S1 S2 D1 D2 JP.WRITE Jn S1 S2 D1 D2 GP.WRITE Un S1 S2 D1 D2	Escribe el dato de la estación anfitrión a la palabra de dispositivo de otras estaciones.		10	—	8-27
	SWRITE	J.SWRITE Jn S1 S2 D1 D2 D3 G.SWRITE Un S1 S2 D1 D2 D3 JP.SWRITE Jn S1 S2 D1 D2 D3 GP.SWRITE Un S1 S2 D1 D2 D3			11	—	8-34
Instrucciones QnA link. Envío de datos	SEND	J.SEND Jn S1 S2 D1 G.SEND Un S1 S2 D1 JP.SEND Jn S1 S2 D1 GP.SEND Un S1 S2 D1	Envía datos (mensajes) a otras estaciones.		8	—	8-41
Instrucciones QnA link Captación de datos	RECV	J.RECV Jn S1 S2 D1 G.RECV Un S1 S2 D1 JP.RECV Jn S1 S2 D1 GP.RECV Un S1 S2 D1	Recibe datos (mensajes) enviados para la estación anfitrión.		8	—	8-4649
Instrucciones QnA link Solicitación momentánea desde otra estación	REQ	J.REQ Jn S1 S2 D1 D2 G.REQ Un S1 S2 D1 D2 JP.REQ Jn S1 S2 D1 D2 GP.REQ Un S1 S2 D1 D2	Envía una solicitación momentánea a otras estaciones y las ejecuta .		8	—	8-55

Tabla 2.36 Instrucciones para Comunicación de Datos (Continuación)

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Instrucciones de QnA link: Lectura de datos desde módulos de instrucción especial en estaciones E/S remotas	ZNFR	JP.ZNFR Jn S1 S2 D1 GP.ZNFR Un S1 S2 D1	Lee datos desde módulos de función especial en estaciones remotas de E/S.		8	—	8-67
Instrucciones de QnA link:: Escritura de datos en módulos de función especial en estaciones remotas E/S	ZNTO	J.ZNTO Jn S1 S2 D JP.ZNTO Jn S1 S2 D G.ZNTO Un S1 S2 D GP.ZNTO Un S1 S2 D	Escribe datos en módulos de función especial en estaciones de E/S remotas.		8	—	8-72
Instrucciones compatibles de A-series link: Lectura de datos de dispositivos de otras estaciones	ZNRD	J.ZNRD Jn n1 D1 S n2 D2 JP.ZNRD Jn n1 D1 S n2 D2	Lectura de palabras de datos de dispositivos de otras estaciones a estaciones anfitrión.		32	—	8-77 8-81
Instrucciones compatibles de A-series link: Escritura de datos de dispositivos a otras estaciones	ZNWR	J.ZNWR Jn n1 D1 S n2 D2 JP.ZNWR Jn n1 D1 S n2 D2	Escritura de datos de estaciones anfitrión a palabras de dispositivos a otras estaciones.		32	—	8-84 8-88
Instrucciones compatibles de A-series link: Lectura de datos desde módulos de función especial en estaciones remotas de E/S	RFRP	G.RFRP Un n1 D1 n2 D2 GP.RFRP Un n1 D1 n2 D2	Lee datos desde módulos de función especial en estaciones remotas de E/S.		11	—	8-91
Instrucciones compatibles de A-series link: Escritura de datos para módulos de función especial en estaciones remotas de E/S	RTOP	G.RTOP Un n1 D1 n2 D GP.RTOP Un n1 D1 n2 D	Escribe datos para módulos de función especial en estaciones remotas de E/S.		11	—	8-95
Información de enrutamiento de lectura	RTREAD	Z.RTREAD n D ZP.RTREAD n D	Lee datos activos en parámetros de enrutamiento		7	—	8-99
Información de registro de enrutamiento	RTWRITE	Z.RTWRITE n S ZP.RTWRITE n S	Escribe datos routing en el área designada por los parámetros de enrutamiento		8	—	8-103

2.5.20 Instrucciones de QCPU

Tabla 2.37 Instrucciones de QCPU

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Lectura de información de módulos	UNIRD	UNIRD n1 D n2	• Lee la información del modulo almacenada desde el área del número de E/S designada por (n) mediante los puntos designados por (n2), y lo almacena en el área empezando por el dispositivo designado por (d).	⎍	4	—	9-2
	UNIRDP	UNIRDP n1 D n2		⎍			
Activación de Trazado	TRACE	TRACE	• Almacena datos de trazado al periférico para guardar en memoria mediante el número designado cuando SM800, SM801, y SM802 se encuentran activas.	⎍	1	—	9-6
Reset de Trazado	TRACER	TRACER	• Resetea trazado	⎍	1	—	9-6
Escritura de datos en archivos designados	SP.FWRITE	SP.FWRITE U0 S0 D0 S1 S2 D1	• Escribe datos en archivos designados.	⎍	11	—	9-8
Lectura de datos desde archivos designados	SP.FREAD	SP.FREAD U0 S0 D0 S1 S2 D1	• Lectura de datos desde archivos designados.	⎍	11	—	9-16
Carga de programa en memoria	PLOADP	PLOADP S D	• Transfiere el programa almacenado en la memoria (unidad diferente de 0) hacia el registro 0 y almacena el programa en estado de espera.	⎍	3	—	9-27
Descarga de programa desde memoria de programa	PUNLOADP	PUNLOADP S D	• Borra el programa en espera almacenado en la memoria de la unidad 0.	⎍	3	—	9-30
Carga + Descarga	PSWAPP	PSWAPP S1 S2 D	• Borra el programa en espera almacenado en memoria en la unidad 0 designado por (S1). Posteriormente, transfiere el programa almacenado en memoria diferente a la unidad 0 designada por (S2) a la unidad 0 y lo mantiene en estado de espera.	⎍	4	—	9-33
Transferencia de bloques de registros de archivo en alta velocidad	RBMOV	RBMOV S D n	• Transfiere n puntos de 16 bits de datos desde el dispositivo designado por (S) hacia los dispositivos empezando correlativamente por el dispositivo designado por (D).	⎍	4	—	9-36
	RBMOVP	RBMOVP S D n		⎍			
Escritura en posición de memoria en CPU de estación anfitrión	S. TO	S.TO n1 n2 n3 n4 D	• Escribe el dato del dispositivo de la estación anfitrión en la posición de la memoria del módulo de CPU de la estación anfitrión.	⎍	5	—	9-41
	SP. TO	SP.TO n1 n2 n3 n4 D		⎍			
	TO	TO n1 n2 Ⓢ n3	• Escribe el dato del dispositivo de la estación anfitrión en la posición de memoria de la CPU del módulo de CPU de la estación anfitrión.	⎍	5	—	9-44
	TOP	TOP n1 n2 Ⓢ n3		⎍			
Lectura posición de memoria desde otra estación CPU	FROM	FROM n1 n2 D n3	• Lee el dato del dispositivo desde el área de memoria de otra CPU de otro módulo CPU de otra estación a la estación anfitrión.	⎍	5	—	9-48
	FROMP	FROMP n1 n2 D n3		⎍			

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Refresco automático de posición de memoria de la CPU	COM	COM	• Realiza el refresco automático de el modulo de funciones especiales, dato general de proceso, y el automático refresco de la posición de memoria de la CPU.		1	—	9-50

2.5.21 Instrucciones de sistema redundantes (Para Q4ARCPU)

Tabla 2.38 Instrucciones de sistema redundantes (Para Q4ARCPU)

Categoría	Nomenclatura	Símbolo	Detalles Procesamiento	Condición de Ejecución	Número de pasos	Subprograma	Ver manual de programación
Operaciones de puesta en modo durante la inicialización de la CPU	S.STMODE	S.STMODE S1 S2	• Designa el modo de operación en (S1) si se quiere borrar el dispositivo Q4ARCPU antes de inicializarlo o no cuando la fuente de alimentación ha sido habilitada en la inicialización de la CPU.		9	—	10-2
Operaciones de puesta en modo durante la conmutación de la CPU	S.CGMODE	S.CGMODE S	• Designa el modo de operación en (S1) si se quiere borrar el dispositivo Q4ARCPU antes de inicializarlo o no cuando la fuente de alimentación ha sido conmutada desde el sistema de control.		7	—	10-4
Seguimiento de Datos	S.TRUCK	S.TRUCK S	• Realiza un seguimiento de datos de la memoria de acuerdo con los parámetros de bloques de datos contenidos en el área de almacenamiento empezando desde el dispositivo designado por (S) durante la ejecución de la instrucción END.		6	—	10-6
Refresco de lotes de memoria del buffer	S.SPREF	S.SPREF S	• Escribe/Lee por lotes los buffers de memoria de los módulos de funciones especiales de acuerdo con el contenido de los parámetros de los bloques de datos almacenados en el área apuntada por el dispositivo designado por (S).		6	—	10-10

3. INSTRUCCIONES DE CONFIGURACIÓN

3.1 Instrucciones de configuración

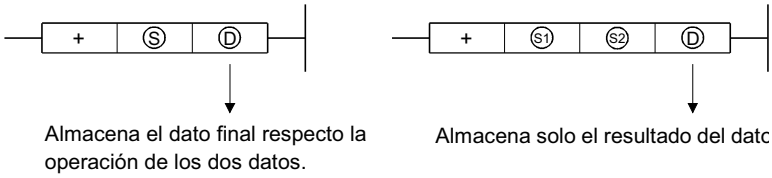
Muchas instrucciones para módulos de CPU consisten en una parte de instrucción y otra parte de datos.

• Parte de instrucción....Indica la función de la instrucción.
• Parte de datos.....Indica el dato que va a ser usado en la instrucción.

La parte de dispositivo se clasifica en fuente, dato de destino y número de dispositivos.

(1) Fuente Ⓢ
(a) Fuente es el dato usado para operaciones.
(b) Los siguientes tipos de fuentes son accesibles, dependiendo de los dispositivos designados:
• Constantes .....Designa el valor numérico a ser usado en la operación.
Se define cuando se escribe el programa, y no puede ser cambiado durante la ejecución del programa..
Las constantes deben ser indexadas cuando son usadas como dato variable.
• Datos de Bit y datos Word.....Designan el dispositivo que almacena el dato para ser usado para la operación.
El dato debe ser almacenado en el dispositivo designado hasta que la operación es ejecutada..
Mediante el cambio del dato almacenado en el dispositivo designado durante la ejecución del programa, el dato a ser usado en la instrucción puede ser cambiado.

(2) Destino Ⓓ
(a) El destino de almacenamiento del dato después de la operación ha sido realizado.
Por otra parte, algunas instrucciones requieren acumulación de datos para posteriores operaciones.
Ejemplo: La instrucción de suma mediante datos BIN de 16 bits

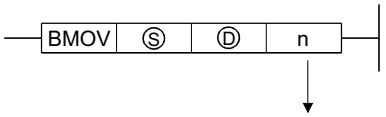


(b) Un dispositivo para el almacenamiento de datos debe siempre ser habilitado para el destino.

(3) Número de dispositivo y número de transferencia(n)

(a) El número de dispositivo y número de transferencia designan el número de dispositivo y transferencia usados por instrucciones implicando dispositivos múltiples.

Ejemplo: Instrucciones de transferencia de bloques

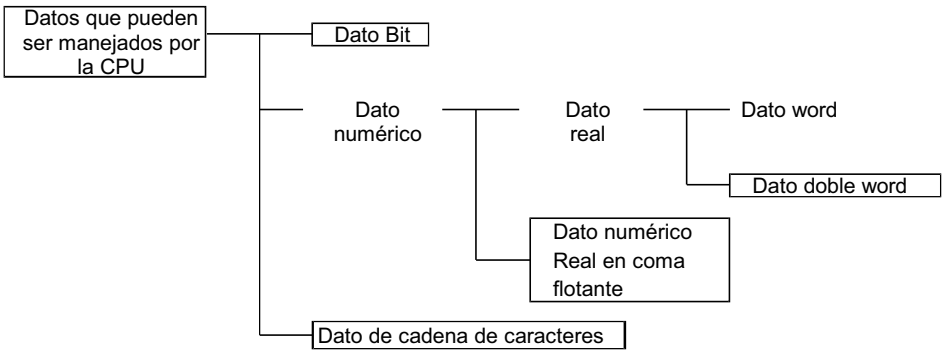


Designa el número de transferencias
Usadas por la instrucción BMOV

(b) El número de dispositivos o número de transferencias pueden ser habilitadas entre 0 y 32767.
Pero, si el número es 0, la instrucción será la instrucción de no funcionamiento.

3.2 Denominación de datos

Los siguientes 5 tipos de datos pueden ser usados con instrucciones referentes a módulos CPU:

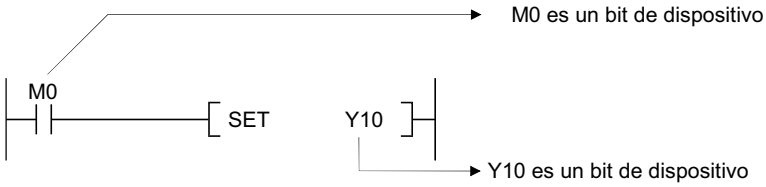


3.2.1 Marcas bit

El dato Bit es un dato utilizado en unidades de un bit, tales como contactos de puntos o bobines. “Dispositivo de Bit” y “Dispositivos de palabra designadas como Bit ” pueden ser usadas como bits de datos.

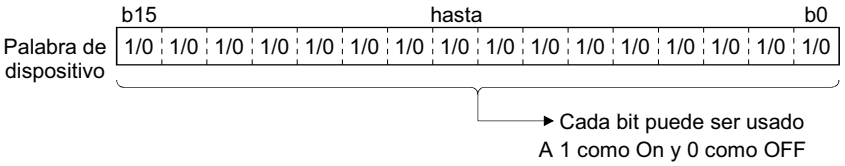
(1) Cuando se utilizan bits de datos

Los dispositivos de bits son designados en unidades de un punto.

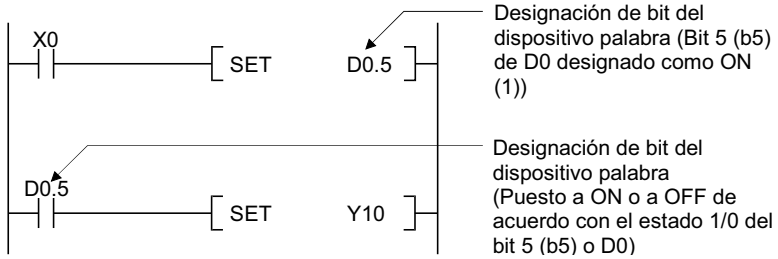


(2) Usando dispositivo de palabras

(a) Las palabras de dispositivo habilitan el uso de número de bits designados 1/0 como bits de datos por la designación de susodicho número de bit..



(b) La designación de bit de palabra de dispositivo se realiza por la designación "Palabra de dispositivo Núm. Bit".
(Designación de número de bits realizada en hexadecimal.)
Por ejemplo, bit 5 (b5) de D0 se designa como D0.5, y el bit 10 (b10) de D0 se designa como D0.A.
Pero, no puede haber designación de bits de temporizadores (T), temporizadores retentivos (ST), contadores (C) o registros índice (Z). (en el ejemplo el registro Z0.0 no es aplicable)


X0
SET
D0.5
Designación de bit del dispositivo palabra (Bit 5 (b5) de D0 designado como ON (1))
D0.5
SET
Y10
Designación de bit del dispositivo palabra (Puesto a ON o a OFF de acuerdo con el estado 1/0 del bit 5 (b5) o D0)

3.2.2 Registros de 16 bits

La palabra de datos es un dato numérico de 16 bits usado con instrucciones básicas e instrucciones de aplicación.
Los siguientes dos tipos de palabras pueden ser usados con los módulos de CPU:
• Constantes decimales.....K-32768 hasta K32767
• Constantes Hexadecimales .....H0000 hasta HFFFF

Dispositivos de palabra y dispositivos designados por dígitos pueden ser usados como palabra de datos.
Para entradas de acceso directo (DX) y salidas de acceso directo (DY), la palabra de dato no puede ser designada por el dígito de salida (DY). (Para más detalles de entradas de acceso directo y salidas de acceso directo, consultar el manual de usuario (Explicaciones de funciones, programa principal) del módulo en uso de la CPU, o el manual de programación básico de QnACPU).
(1) Cuando se usan dispositivos bit
(a) Dispositivo de bit puede usarse con palabras de datos cuando los dígitos están designados.
La designación de dígitos de dispositivo de bits se realiza mediante la designación " Numero de dígitos Número inicial de bit de dispositivo ". La designación de bits de dispositivos puede ser realizada en 4 unidades de puntos (4-bit) y la designación puede ser hecha entre K1 y K4.
(Para dispositivos directos de comunicación, la designación se realiza mediante "J Núm.Red \ Designación de dígito Número inicial de dispositivos bit ". Cuando de X100 hasta X10F son designados por la red Número.2, se realiza mediante J2\K4X100.)
Por ejemplo, si X0 se nombra mediante la designación del dígito, los siguientes números serían designados:
• K1X0.....Los 4 puntos X0 hasta X3 son designados
• K2X0.....Los 8 puntos X0 hasta X7 son designados
• K3X0.....Los 12 puntos X0 hasta XB son designados
• K4X0.....Los 16 puntos X0 hasta XF son designados

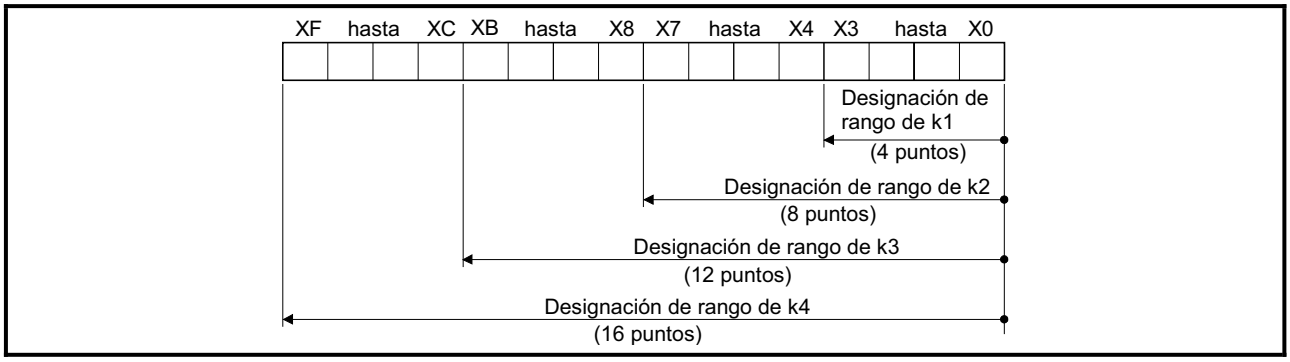

XF hasta XC XB hasta X8 X7 hasta X4 X3 hasta X0
Designación de rango de k1
(4 puntos)
Designación de rango de k2
(8 puntos)
Designación de rango de k3
(12 puntos)
Designación de rango de k4
(16 puntos)

Fig. 3.1 Direccionamiento de dígitos de posición de rango para instrucciones de 16 bits

(b) En casos donde la direccionamiento del dígito ha sido realizada en la fuente, los valores numéricos mostrados en la tabla 3.1 son aquellos que pueden ser usados como datos fuente.

Tabla 3.1 Lista de valores numéricos que pueden ser usados con direccionamiento de dígitos

Número de direccionamiento de dígitos	Con instrucciones de 16 bits
K1 (4 puntos)	0 a 15
K2 (8 puntos)	0 a 255
K3 (12 puntos)	0 a 4095
K4 (16 puntos)	-32768 a 32767

En casos donde la fuente es un bit de dispositivo designado por el direccionamiento del dígito y la designación es una palabra de dispositivo, la palabra de dispositivo para el destino pasa a ser 0 seguida del bit designado por el direccionamiento del dígito en la fuente.

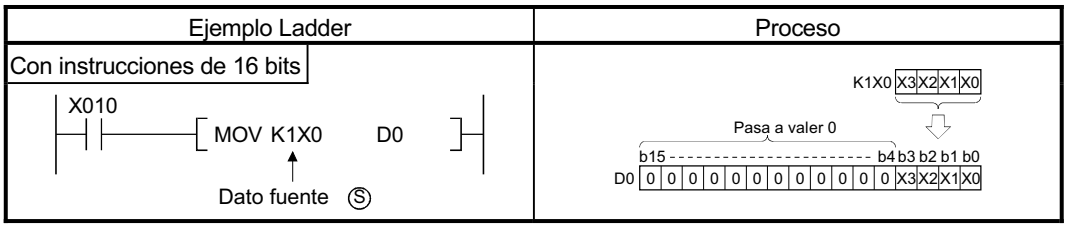

Ejemplo Ladder
Proceso
Con instrucciones de 16 bits
X010
MOV K1X0 D0
Dato fuente (S)
K1X0 X3 X2 X1 X0
Pasa a valer 0
b15 - b4 b3 b2 b1 b0
D0 0 0 0 0 0 0 0 0 0 0 0 X3 X2 X1 X0

Fig. 3.2 Ejemplo Ladder y ejecución de proceso

(c) En casos donde el direccionamiento del dígito se realiza en el destino, el número de puntos designados es utilizado como el destino. Los bits de dispositivo seguidos del número de puntos designados como dígitos no cambian.

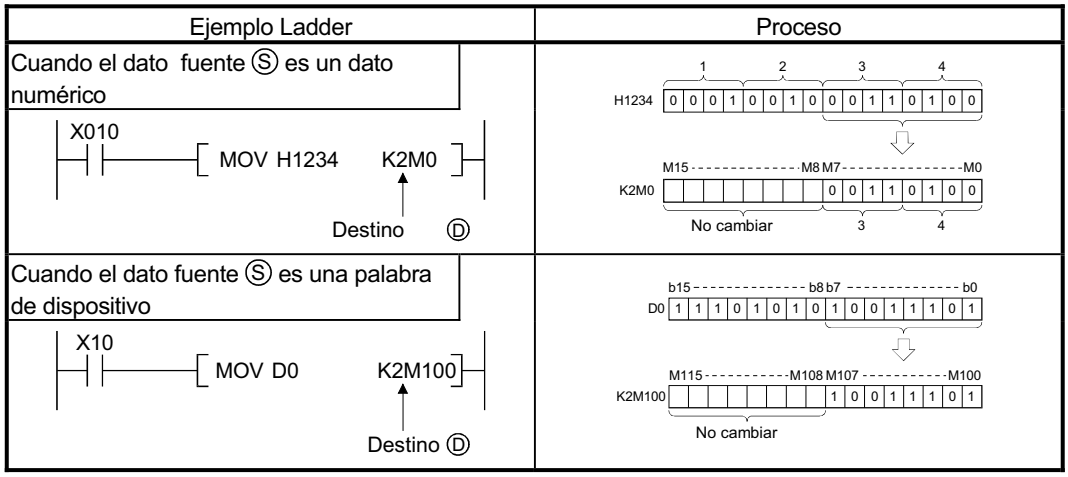
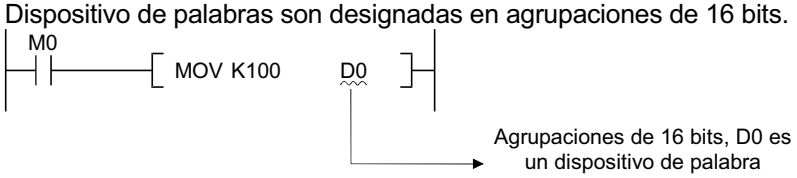

Ejemplo Ladder
Proceso
Cuando el dato fuente Ⓢ es un dato numérico
X010
MOV H1234 K2M0
Destino Ⓓ
1 2 3 4
H1234 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0
M15 M8 M7 M0
K2M0 0 0 1 1 0 1 0 0
No cambiar 3 4
Cuando el dato fuente Ⓢ es una palabra de dispositivo
X10
MOV D0 K2M100
Destino Ⓓ
b15 b8 b7 b0
D0 1 1 1 0 1 0 1 0 1 0 0 1 1 1 0 1
M115 M108 M107 M100
K2M100 1 0 0 1 1 1 0 1
No cambiar

Fig. 3.3 Ejemplo Ladder y proceso

(2) Cuando usamos dispositivos de palabra

Dispositivo de palabras son designadas en agrupaciones de 16 bits.



Agrupaciones

(1) Cuando se realizan procesos de designación de dígitos, un valor aleatorio puede ser usado por el número de dispositivo del bit de dispositivo inicial.

(2) La designación del dígito no puede ser realizada por el acceso directo E/S (DX, DY).

3.2.3 Uso de datos de doble palabra (32 bits)

Los datos de "double word" (doble palabra) son registros de datos numéricos de 32 bits usados por instrucciones básicas e instrucciones de aplicación.
Los dos tipos de datos doble palabra que pueden ser tratados por módulos CPU son los siguientes:

• Constantes decimales.....K-2147483648 hasta K2147483647
• Constantes hexadecimales…H00000000 hasta HFFFFFFFF

Dispositivos de palabras y dispositivos designados por direccionamiento de dígitos pueden ser utilizados como datos de doble palabra(32 bits).
Para entradas de acceso directo (DX) y salidas de acceso directo (DY), la asignación de datos dobles de palabra no es posible por el direccionamiento del dígito.

(1) Cuando usamos dispositivos de bits

(a) El direccionamiento de dígitos puede ser utilizado para habilitar el bit de dispositivo y tratar con datos de doble palabra. El direccionamiento de dígitos de dispositivos bit se realiza por la designación "Numero de dígitos \ Número inicial del bit del dispositivo".
El direccionamiento de dígitos de dispositivos bit puede hacerse en agrupaciones de 4 bits, y el direccionamiento puede hacerse de K1 hasta K8.
(Para dispositivos de comunicación directa, el direccionamiento se realiza mediante "J Núm. ref. \ Digito \ designación
Valor inicial del bit de dispositivo". Cuando de X100 hasta X11F son designados por la red Núm.2, se realiza mediante J2\K8X100.)
Por ejemplo, si X0 se designa por direccionamiento de dígitos, las siguientes agrupaciones serian las designadas por:

• K1X0.....Las agrupaciones de 4 bits X0 hasta X3 serían designadas
• K2X0.....Las agrupaciones de 8 bits X0 hasta X7 serían designadas
• K3X0.....Las agrupaciones de 12 bits X0 hasta XB serían designadas
• K4X0.....Las agrupaciones de 16 bits X0 hasta XF serían designadas
• K5X0.....Las agrupaciones de 20 bits X0 hasta X13 serían designadas
• K6X0.....Las agrupaciones de 24 bits X0 hasta X17 serían designadas
• K7X0.....Las agrupaciones de 28 bits X0 hasta X1B serían designadas
• K8X0.....Las agrupaciones de 32 bits X0 hasta X1F serían designadas

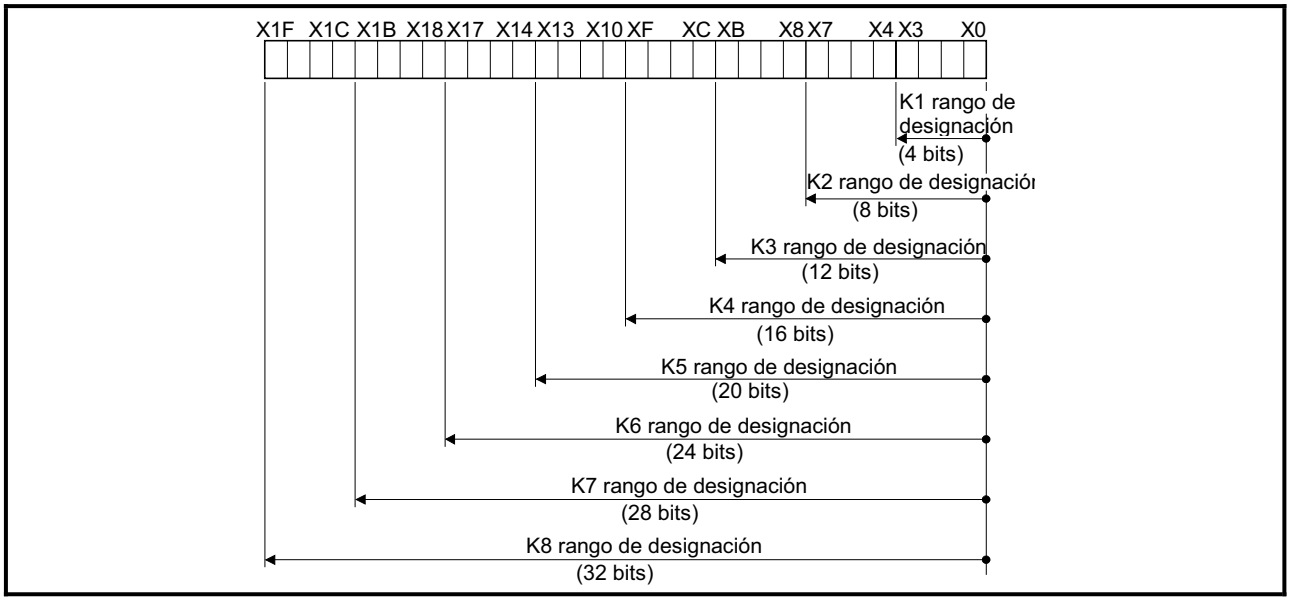

X1F X1C X1B X18 X17 X14 X13 X10 XF XC XB X8 X7 X4 X3 X0
K1 rango de designación (4 bits)
K2 rango de designación (8 bits)
K3 rango de designación (12 bits)
K4 rango de designación (16 bits)
K5 rango de designación (20 bits)
K6 rango de designación (24 bits)
K7 rango de designación (28 bits)
K8 rango de designación (32 bits)

Fig. 3.4 Rango de Direccionamiento de dígitos para instrucciones de 32 bits

(b) En casos donde el direccionamiento de dígitos ha sido realizado en la fuente Ⓢ, los valores numéricos mostrados en la Tabla 3.2 son aquellos que pueden ser tratados como datos fuente.

Tabla 3.2 Lista de valores numéricos que pueden ser tratados como direccionamiento de dígitos.

Número de dígitos direccionados	Con instrucciones de 32 bits	Número de dígitos designados	Con instrucciones de 32 bits
K1 (4 bits)	0 hasta 15	K5 (20 bits)	0 hasta 1048575
K2 (8 bits)	0 hasta 255	K6 (24 bits)	0 hasta 16777215
K3 (12 bits)	0 hasta 4095	K7 (28 bits)	0 hasta 268435455
K4 (16 bits)	0 hasta 65535	K8 (32 bits)	-2147483648 hasta 2147483647

En casos donde la fuente es un bit de dispositivo designado por direccionamiento de dígito, y el destino es una palabra de dispositivo, la palabra de dispositivo para el destino pasa a valer 0 seguido del bit designado por el dígito designado en la fuente.

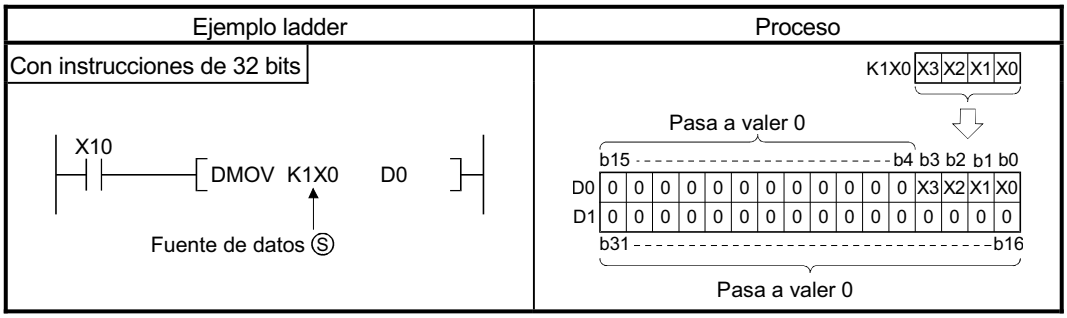

Ejemplo ladder
Proceso
Con instrucciones de 32 bits
X10
DMOV K1X0 D0
Fuente de datos Ⓢ
K1X0 X3 X2 X1 X0
Pasa a valer 0
b15 b4 b3 b2 b1 b0
D0 0 0 0 0 0 0 0 0 0 0 0 X3 X2 X1 X0
D1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
b31 b16
Pasa a valer 0

Fig. 3.5 Ejemplo ladder y ejecución de proceso

(c) En casos donde la designación del dígito ha sido realizada en la fuente Ⓓ, los valores numéricos mostrados en la Tabla 3.2 son aquellos que pueden ser tratados como un dato fuente.
Los bits de dispositivo después del número de bits designados como dígitos no cambian.

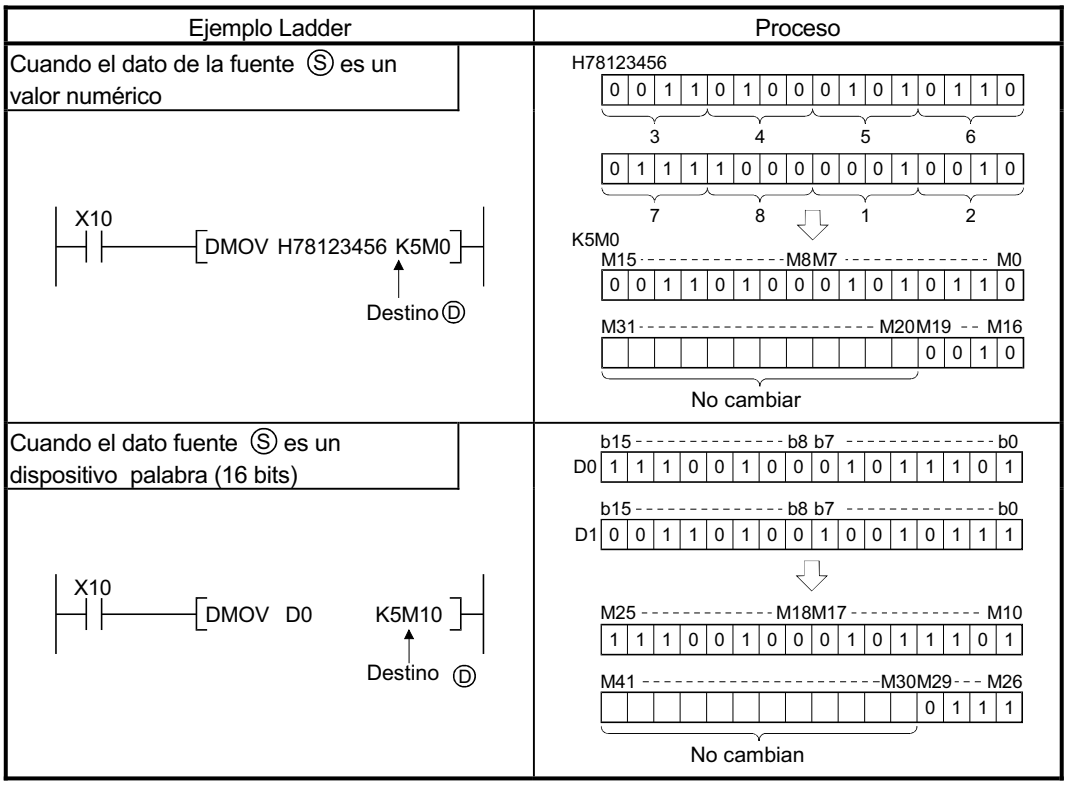


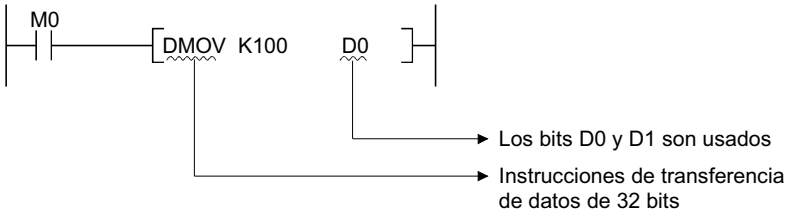
Fig. 3.6 Ejemplo Ladder y ejecución de proceso

Comentarios

(1) Cuando el dígito de designación de proceso es ejecutado, un valor aleatorio puede ser usado por el dispositivo de bit del número inicial de dispositivo.
(2) La designación del dígito no puede ser utilizada para el acceso directo I/O (DX, DY).

(2) Cuando se usan dispositivos palabra

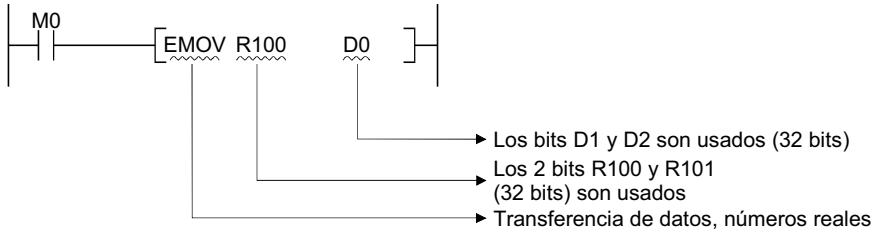
Un dispositivo de palabra designa dispositivos usados por los 16 bits más bajos de datos.
Una instrucción de 32 bits utiliza(número de dispositivo designado) y (número de dispositivo designado + 1).



3.2.4 Uso de datos numéricos reales

Los datos de números reales son datos de 32 bits en coma flotante usados con instrucciones básicas y de instrucciones de aplicaciones.
Sólo los dispositivos de palabra son capaces de almacenar datos de números reales.

Las instrucciones que tratan con números reales designan dispositivos que son usados para los datos de 16 bits más bajos de datos. Los números reales son almacenados en los 32 bits que lo forman (en el número de dispositivo designado) y (el número de dispositivo designado + 1).

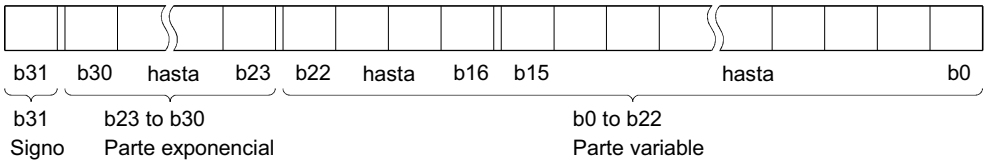

M0
EMOV R100 D0
Los bits D1 y D2 son usados (32 bits)
Los 2 bits R100 y R101
(32 bits) son usados
Transferencia de datos, números reales

COMENTARIOS

1) En la secuencia de programa, los números reales son designados por E[ ].

Los Datos decimales en coma flotante usan dos dispositivos de palabra y se expresan de la siguiente forma:
[Signo] 1. [Parte de la variable] × 2 [Parte del exponente]

El bit de configuración y el significado de la representación interna del dato en coma flotante se representa de la siguiente forma:



• Signo — El signo es representado en b31.
0: Positivo
1: Negativo

• Parte exponencial — El dato n de 2^n se representa desde b23 hasta b30.
Dependiendo del valor BIN de b23 hasta b30, el valor de n es:

b23 hasta b30	FFH	FEH	FDH		81H	80H	7FH	7EH		02H	01H	00H
n	No numérico	127	126		2	1	0	-1		-125	-126	No numérico

• Parte variable — Los 32 bits desde b0 hasta b22, representa el XXXXXX... en binario 1.XXXXXX....

Observaciones

(1) El modulo de CPU de datos decimales en coma flotante pueden ser monitorizados usando la función de monitorización de un dispositivo periférico.

(2) Cuando un dato en coma flotante es usado para expresar 0, todos los datos desde b0 hasta b31 son puestos a cero.

(3) El rango de inicialización de números reales es 0 y $\pm 2^{-126} \leq |$ valor $| < 2^{128}$.

(4) No especificar –0 en datos en coma flotante (solo cuando el bit más significativo del número real en coma flotante es 1). (Ocurrirá un error si la operación en coma flotante es representada por –0.
El modulo CPU que efectúa la operación interna de la operación puesto que convierte –0 a 0 en la CPU cuando se especifica –0.
El modulo CPU que representa la operación interna de la operación en coma flotante con precisión simple muestra un error en la operación, puesto que da prioridad a la velocidad de proceso y usa –0 en operación en coma flotante sin conversión cuando se especifica –0 .

(a) Los siguientes módulos de CPU no darán error cuando se especifica –0.
Los modelos QCPU altas prestaciones dónde las operaciones internas son inicializadas en precisión doble ∗1
(La operación interna de operación en coma flotante por defecto es de doble precisión.)
- QnACPU
- Q2ASCPU
- Q4ARCPU cuyo contacto SM707 es puesto a ON

(b) Los siguientes módulos de CPU darán error cuando se especifique –0 .
- Modelos básicos de QCPU ∗2
- Modelos QCPU de altas prestaciones donde la operación interna es inicializada en precisión simple∗1
- Proceso CPU
- Q4ARCPU cuyo contacto SM707 es puesto a OFF

Comentarios

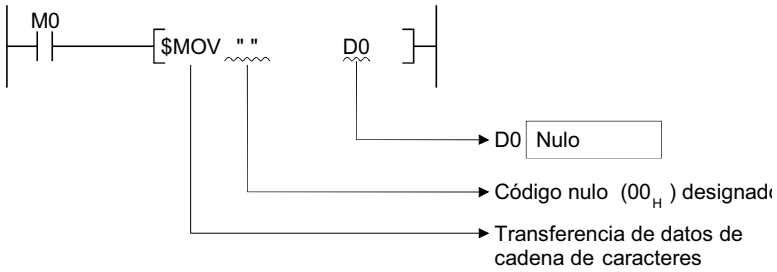
∗1: Conmute entre precisión simple y doble de la operación interna de funcionamiento en coma flotante en el sistema PLC, ajustando los parámetros de la ventana de diálogo de parámetros del PLC.
Para la precisión simple y doble de funcionamiento en coma flotante, refiérase a modelos QCPU de altas prestaciones (modo Q) Manual de usuario (Explicación de funciones, Fundamentos de programación.

∗2: Los modelos QCPU básicos pueden ejecutar operaciones en coma flotante si sus cinco primeros dígitos del número de serie son "04122" o superiores.

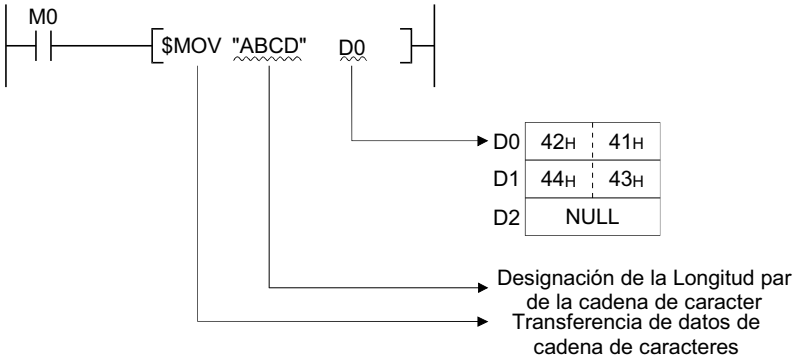
3.2.5 Uso de datos de cadenas de caracteres

Un dato de cadenas de caracteres es un dato carácter usado por instrucciones básicas y instrucciones de aplicación. Abarca todos los datos desde el carácter designado hasta el código nulo 00_H.

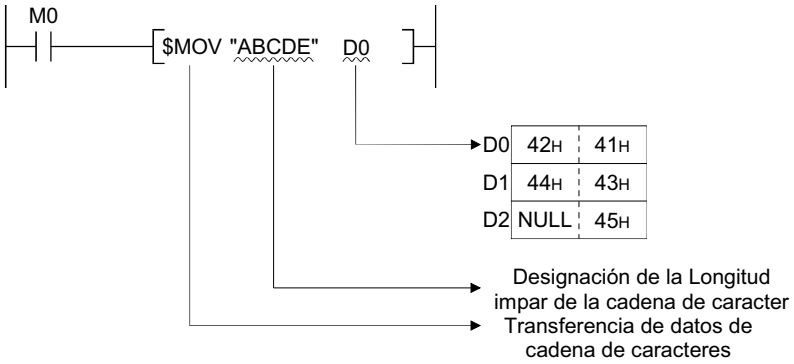
(1) Cuando el carácter designado es el código nulo.
Una palabra es usada para almacenar el código nulo.



(2) Cuando la longitud de la cadena de caracteres es par
Usa palabras (Números de Caracteres /2 + 1) y almacena la cadena de caracteres y el código NULL.
Por ejemplo, si "ABCD" se transfiere a D0, la cadena de caracteres ABCD se almacena en D0 y D1 y el código NULL es almacenado en D2.



(3) Cuando el número de caracteres es impar
Usa palabras(Número de caracteres/2) (redondea fracciones decimales) y almacena la cadena de caracteres y el código NULL.
Por ejemplo si "ABCDE" es transferido a D0, la cadena de caracteres (ABCDE) y los códigos NULL son almacenados de D0 a D2.



3.3 Modificación de Índice

(1) Modificación de Índice

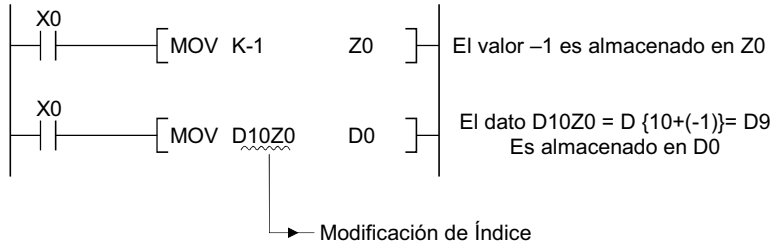
(a) La modificación de índice es una inicialización indirecta hecha usando un registro índice. Cuando un índice de modificación se usa en una secuencia de programa, el dispositivo será el número de dispositivo designado directamente sumado el contenido del registro índice.

Por ejemplo, si D2Z2 ha sido designado, la designación de dispositivo será calculada tal y como se especifica:

D(2+3)=D5 y el contenido de Z2 es 3 pasa a valer el dispositivo designado.

(b) Hay 16 registros índices, desde Z0 hasta Z15.

Cada registro índice puede ser inicializado entre -32768 y 32767.



Ejemplo

En un caso donde la modificación del índice ha sido representada y el actual dispositivo de proceso, sería como el siguiente: (Cuando Z0 = 20 y Z1 = -5)

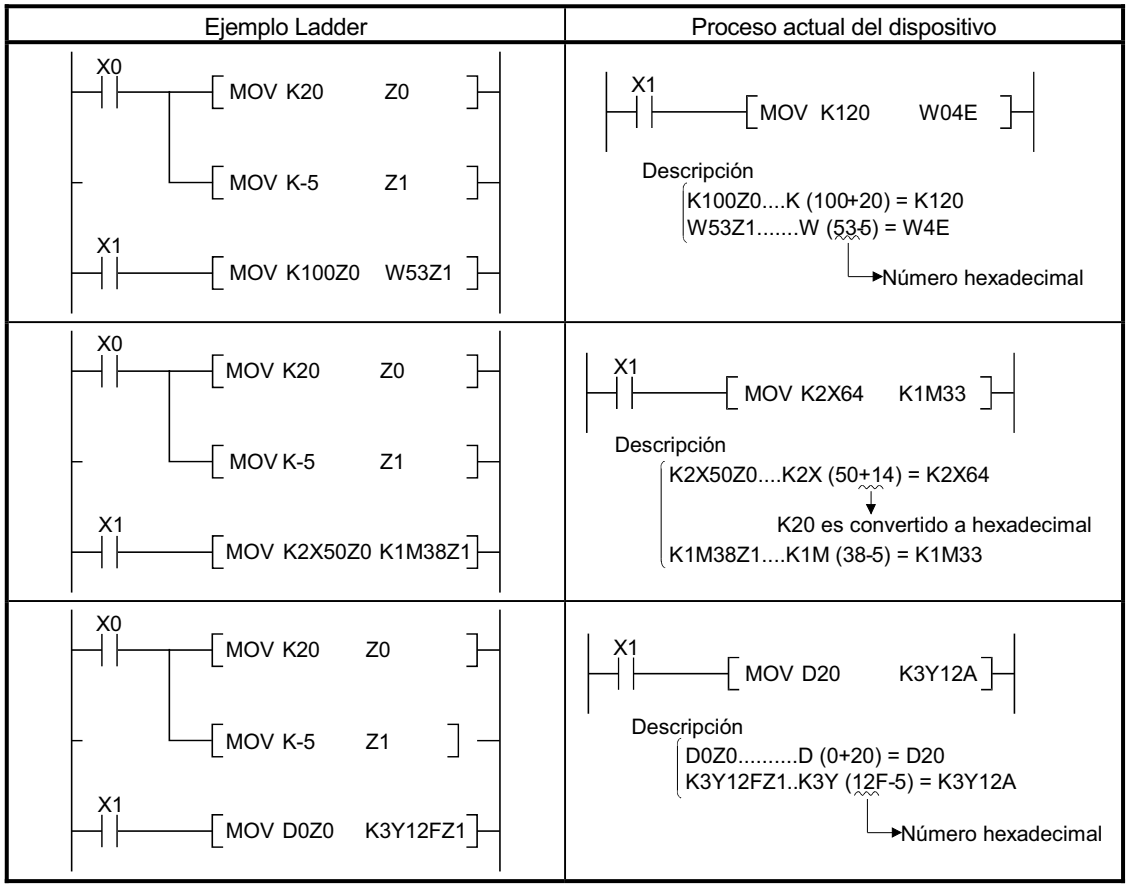


Fig. 3.7 Ejemplo Ladder y proceso actual de dispositivo

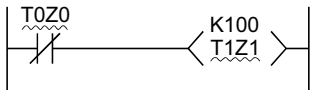
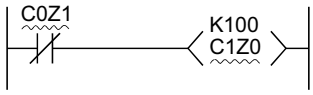
(2) Dispositivos que pueden ser modificados por índice.

Con la excepción de las restricciones anotadas abajo, la modificación del índice puede ser usada con dispositivos usados con contactos, bobinas, instrucciones básicas y aplicaciones de instrucciones.

(a) Dispositivos que no pueden ser modificados por índice

Dispositivo	Significado
K, H	Constante de 32 bits
E	Dato decimal en coma flotante
$	Dato de cadena de caracteres
⬚, ⬚	Bit designado para dispositivo de palabra
FX, FY, FD	Dispositivo de funciones
P	Apuntadores usados como etiquetas
I	Punteros de interrupciones usados como etiquetas
Z	Registros índice
S	Pasos de relé
TR	Dispositivo de transferencia GRAFCET ∗1
BL	Dispositivo de bloques GRAFCET ∗1
T, ST	Valor puesto por temporizador
C	Valor puesto por contador

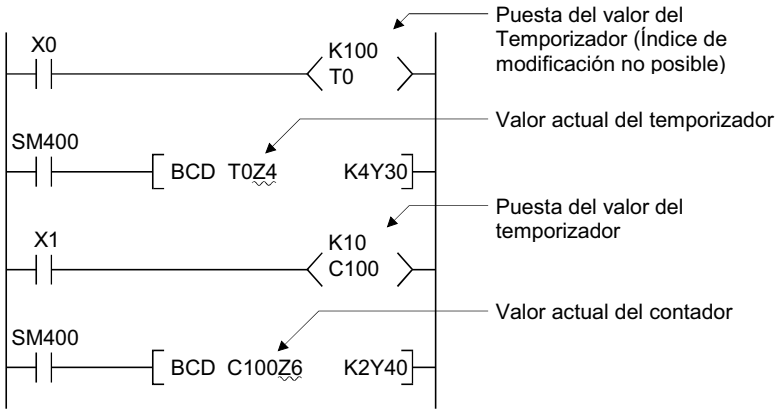
(b) Dispositivos con limitaciones para usar con índices de registro

Dispositivo	Significado	Ejemplo de modificación
T	• Solo Z0 y Z1 puede ser usado por contactos temporizados y bobinas	
C	• Solo Z0 y Z1 puede ser usado por contactos de contadores y bobinas	

Observaciones

1) ∗1: Los dispositivos de transferencia GRAFCET y bloques de dispositivo GRAFCET son dispositivos para uso de GRACFET.
Referirse al manual de programa QCPU (modo Q)/QnACPU (SFC) para información de cómo usar estos dispositivos.

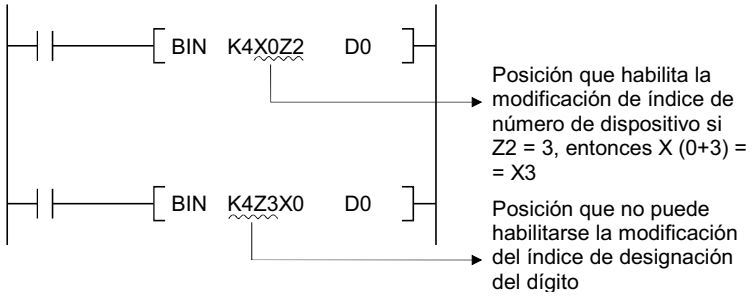
2) Para valores actuales de temporizadores y contadores, no hay límites de número índice de registros usados.



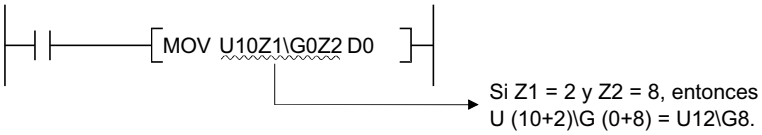
(c) Otros

1) Datos Bit

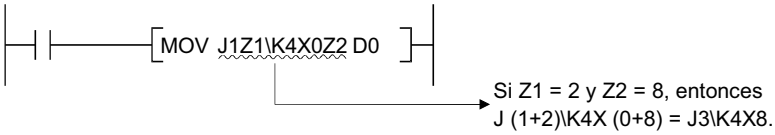
Número de dispositivo que puede ser modificado en índice cuando se representa la designación del dígito.


BIN K4X0Z2 D0
Posición que habilita la modificación de índice de número de dispositivo si Z2 = 3, entonces X (0+3) = = X3
BIN K4Z3X0 D0
Posición que no puede habilitarse la modificación del índice de designación del dígito

2) Ambas posiciones E/S y buffer de memoria pueden ser modificadas por índice con la función especial de módulo de dispositivo.



3) Ambos números de redes y número de dispositivos pueden ser modificadas por el índice para conexión de dispositivos directos.

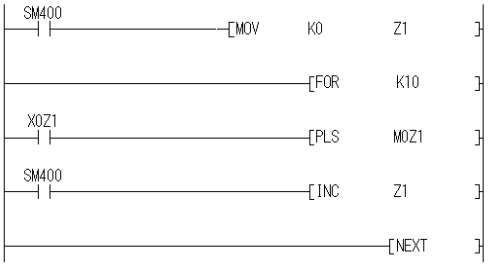


Observaciones

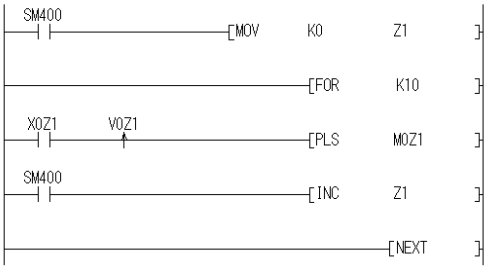
1) ∗1: Referirse al manual de usuario (Explicación de funciones, Programación fundamental) del uso de módulos CPU o manual de programación QnACPU (Fundamental) para dispositivos de módulos de función especial.
2) ∗2: Referirse al manual de usuario (Explicación de funciones, Programación fundamental) del uso de módulos CPU o manual de programación QnACPU (Fundamental) para conexión de dispositivos directos.

(3) Precaución con el tratamiento de modificación de índices entre instrucciones FOR y NEXT.
Cuando se indexa el dispositivo de instrucciones PLS/PLF/ ejecutadas solo durante el flanco ascendente de OFF a ON (□P) entre las instrucciones FOR y NEXT, usar el flanco del bit (V) como una condición de ejecución.
Si el dispositivo de la instrucción PLS/PLF/ ejecutada solo en el flanco ascendente de OFF a ON (□P) entre las instrucciones FOR y NEXT es indexado sin el uso del flanco de bit, las instrucciones PLS/PLF/ ejecutadas solo en el flanco ascendente de OFF a ON (□P) no operarán de forma normal.

[Cuando el flanco de bit no se utiliza]
(PLS M0Z1 no proporciona un pulso de salida normal)



[Cuando el flanco de bit se utiliza]
(PLS M0Z1 proporciona un pulso de salida normal)



Observaciones

∗1: La señal ON/OFF de X0Z1 es almacenada por el flanco del bit V0Z1.
Por ejemplo, la señal ON/OFF de X0 es almacenada por V0 y lo mismo con X1 por V1.

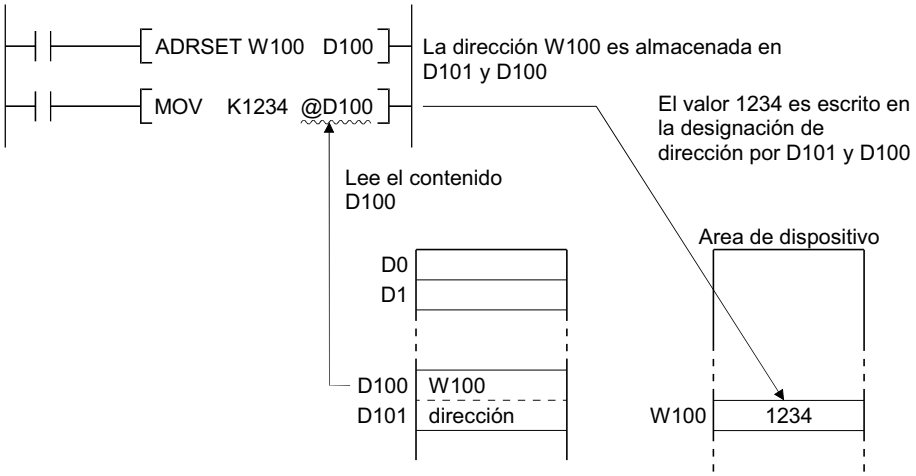
3.4 Designación indirecta

(1) Designación Indirecta

(a) La designación indirecta es una vía de utilización de palabras de dispositivo para designar una dirección de dispositivo que será usada en una secuencia de programa.
Este método puede ser usado cuando el registro índice es insuficiente.

(b) El dispositivo cuya designación de la dirección del dispositivo direccionado es designada por "@+(número de dispositivo de palabra)".
Por ejemplo, el direccionamiento de @D100 hará que el contenido de la dirección del dispositivo D100 D101.

(c) La dirección del dispositivo representada por direccionamiento indirecto puede ser confirmado por el dispositivo de la instrucción ADRSET.


ADRSET W100 D100
La dirección W100 es almacenada en D101 y D100
MOV K1234 @D100
El valor 1234 es escrito en la designación de dirección por D101 y D100
Lee el contenido D100
Area de dispositivo
D0
D1
D100 W100
D101 dirección
W100 1234

(2) Dispositivo admisible de direccionamiento indirecto
Los dispositivos de la CPU que puede ser direccionados indirectamente se muestran en la Tabla 3.3.

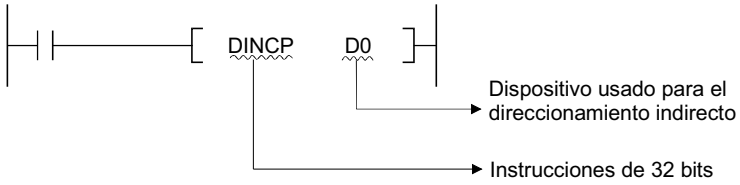
Tabla 3.3 Lista de dispositivos válidos para de direccionamiento indirecto

Tipo de dispositivo		Capaz/Incapaz de direccionamiento indirecto	Ejemplo de direccionamiento indirecto
Dispositivos de uso interno	Dispositivo bit ∗1	No válido	—
	Dispositivo palabra ∗1	Válido	• @D100 • @D100Z2 ∗2
Dispositivo directo de comunicación	Dispositivo bit ∗1	No válido	—
	Dispositivo palabra ∗1	Válido ∗3	• @J1\W10 • @J1Z1\W10Z2 ∗2
Dispositivos directos especiales		Válido ∗3	• @U10\G0 • @U10Z1\G0Z2 ∗2
Registro índice		No válido	—
Registro de archivo		Válido	• @R0, @ZR20000 • @R0Z1, @ZR20000Z1 ∗2
Anidamiento		No Válido	—
Puntero			—
Constantes			—
Otros	Dispositivo de bloques GRAFCET		—
	Dispositivo bajo GRAFCET		
	Número de red		
	Número de E/S		

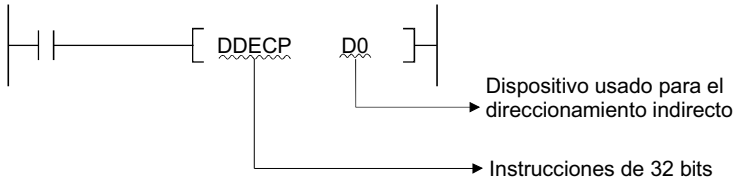
Comentarios

1) ∗1: Referirse al manual de usuario (explicación de funciones, fundamentos de programación) para el uso de módulos de CPU o manuales de programación QnACPU (fundamentos) para nombre de dispositivos.
2) ∗2: Indica la modificación del índice mediante índice de registro
3) ∗3: El dispositivo puede ser designado de forma indirecta, pero la dirección no puede ser escrita en la instrucción de ADRSET.

(3) Precauciones
La dirección para direccionamiento indirecto es designada usando dos palabras.
Por lo tanto, para substituir designaciones indirectas de modificación de índice, se requiere la suma/resta de datos de 32-bit.
Seguidamente se muestra el diagrama ladder usado para la suma/resta de la dirección del dispositivo almacenado en D1 y D0 para la designación indirecta.
[Para sumar “1” a la dirección del dispositivo para direccionamiento indirecto]



[Para restar “1” desde la dirección del dispositivo para direccionamiento indirecto]



3 - 18

3.5 Procesado de subconjunto

El procesado de subconjunto se utiliza para designar límites en los dispositivos de bit usados mediante instrucciones básicas e instrucciones de aplicación para incrementar la velocidad de procesamiento.
Cabe destacar que las instrucciones simbólicas no cambian.
Para abreviar los ciclos de scan, las instrucciones de habilitación se programan según las condiciones indicadas posteriormente.

(1) Condiciones que cada dispositivo tiene que cumplir para el procesado de subconjunto.

(a) Cuando usa un dato de tipo palabra

Dispositivo	Condición
Dispositivo Bit	• Designa un número de dispositivo bit en factor 16 • Solo K4 puede ser designado para el dígito de designación • No afecta la modificación al índice
Dispositivo de palabra	• Dispositivo interno (el archivo de registro ZR no se incluye
Constantes	• Sin limitaciones

(b) Cuando usamos datos de palabras dobles

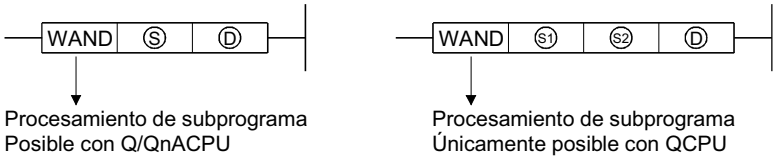
Dispositivo	Condición
Dispositivo Bit	• Designa un número de dispositivo bit en factor 32 • Solo K8 puede ser designado para el dígito de designación • No afecta la modificación al índice
Dispositivo de palabra	• Dispositivo interno (el archivo de registro ZR no se incluye
Constantes	• Sin limitaciones

(2) Instrucciones para las cuales el procesado de subprograma puede ser utilizado

Tipo de instrucciones	Símbolos de las instrucciones
Instrucciones de comparación	• =, < >, <, <=, >, >=, D=, D< >, D<, D<=, D>, D>=
Operaciones aritméticas básicas (suma, resta, multiplicación y división)	• +, -, ＊, /, INC, DEC, D+, D-, D＊, D/, DINC, DDEC • B+, B-, B＊, B/
Instrucciones de conversión de datos	• BCD, BIN, DBCD, DBIN
Instrucciones de transferencia de datos	• MOV, DMOV, CML, DCML, XCH, DXCH • FMOV, BMOV, EMOV (with QCPU only)
Instrucciones de ramificación de programa	• CJ, SCJ, JMP
Operaciones lógicas ＊	• WAND, DAND, WOR, DOR, WXOR, DXOR, WXNR, DXNR
Instrucciones de rotación	• RCL, DRCL, RCR, DRCR, ROL, DROL, ROR, DROR
Instrucciones de desplazamiento	• SFL, DSFL, SFR, DSFR
Instrucciones de proceso de datos	• SUM, SEG
Instrucciones de programa estructurado	• FOR, CALL

Comentarios

1)∗:Solo la familia QCPU puede usar tres dispositivos para efectuar procesando de subconjunto de instrucciones de las operaciones lógicas WAND, DAND, WOR, DOR, WXOR, DXOR, WXNR, o DXNR.



3.6 Precauciones en programación (Errores de funcionamiento)

Los errores de funcionamiento son mostrados por las siguientes causas cuando se ejecutan las instrucciones básicas y las instrucciones de aplicación con el módulo CPU:

- Ha ocurrido un error listado en la página explicativa de la instrucción individual.
- No hay instalado en el número de posición de E/S designada un módulo de función inteligente o de función especial instalado en el número de posición de E/S designada cuando se usa el buffer de registros.
- La red relevante no existe cuando se usa el dispositivo de comunicación.
- Ningún módulo de red se ha instalado en el número de E/S cuando usamos dispositivos de comunicación.

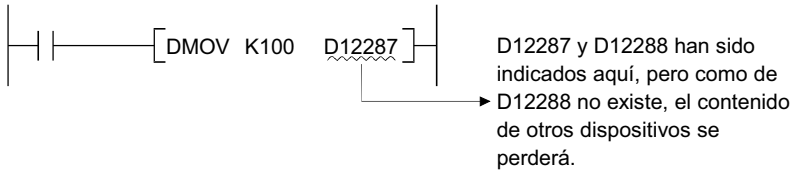
Observaciones

(1) Cuando el ajuste de archivo de registros de habilitación ha sido ejecutado pero no ha sido instalada una tarjeta de memoria, o cuando ningún archivo de registro ha sido realizado, ningún error será devuelto incluso si se intenta escribir en el archivo de registro.
Cabe destacar que el dato "FFFFH" será almacenado si se intenta usar para leer desde un archivo de registros en el cual esta operación de escritura se intentó.

(1) Control de rango de dispositivo

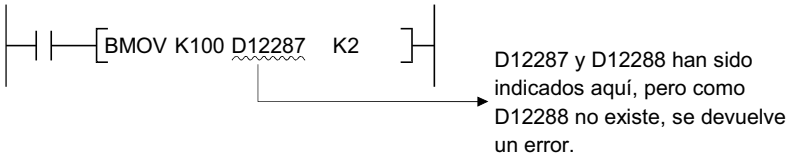
Los controles de rango de dispositivo para el dispositivo usado mediante instrucciones básicas y instrucciones de aplicación en módulos CPU son, como se indican seguidamente:

(a) No se efectúan controles de rango de dispositivo para instrucciones de trato con dispositivos de longitud fija (MOV, DMOV, etc.).
En casos donde el rango de dispositivo correspondiente se excede, el dato es escrito a otros dispositivos. *
Por ejemplo, en casos donde el dato del registro ha sido asignado en 12 k posiciones, no habrá error incluso si excedemos de D12287.


DMOV K100 D12287
D12287 y D12288 han sido indicados aquí, pero como de D12288 no existe, el contenido de otros dispositivos se perderá.

Los controles de rango de dispositivo no se efectúan en casos donde el índice de modificación está siendo representado.

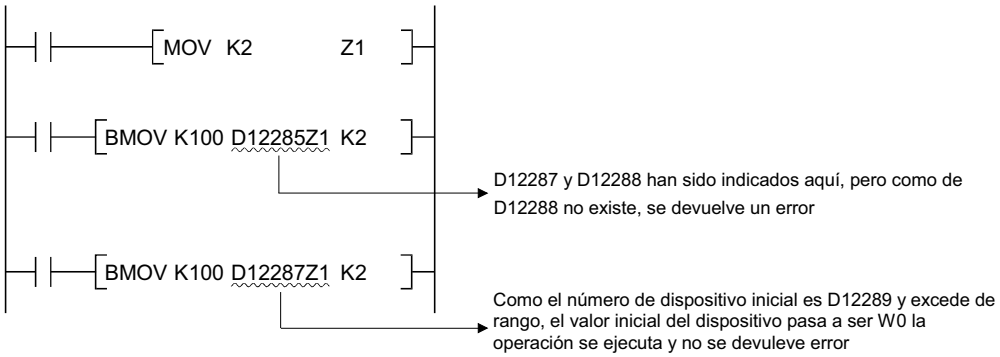
(b) Los controles de rango de dispositivo están dirigidos para instrucciones de trato con dispositivos de longitud variable (BMOV, FMOV y otros que designan números de transferencia).
En casos donde el rango de dispositivo correspondiente ha sido excedido, producirá un error.
Por ejemplo, en casos donde el registro de datos ha sido asignado en 12 k posiciones, habrá un error si se excede de D12287.



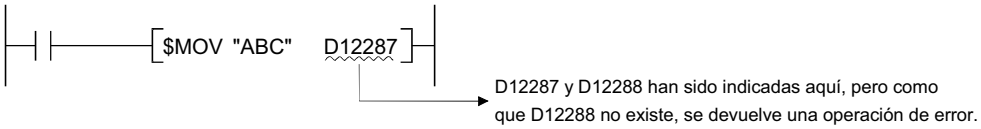
Observaciones

1) ∗: Referirse a la sección 3.4 (3) para el uso interno de orden de asignación de dispositivos.

Los controles de rango de dispositivo están solamente dirigidos cuando el índice de modificación es representado.
Cabe destacar que si el índice de modificación ha sido dirigido, no habrá retorno de error si el número de dispositivo inicial excede del rango de dispositivo pertinente.


MOV K2 Z1
BMOV K100 D12285Z1 K2
D12287 y D12288 han sido indicados aquí, pero como de D12288 no existe, se devuelve un error
BMOV K100 D12287Z1 K2
Como el número de dispositivo inicial es D12289 y excede de rango, el valor inicial del dispositivo pasa a ser W0 la operación se ejecuta y no se devuleve error

(c) Como todos los datos de la cadena de caracteres son de longitud variable, el control del rango de dispositivo se efectúa. En casos donde el rango de dispositivo correspondiente al rango de dispositivo ha sido excedido, se retornará un error.
Por ejemplo, en casos donde el registro de datos ha sido asignado a 12 k posiciones, habrá un error si se excede de D12287.



Notar que una operación de error no ocurre incluso si el valor del dispositivo más relevante excede del rango del dispositivo como resultado de una modificación del índice.

(d) Los controles de rango se ejecutan cuando el índice de modificación es representado mediante la salida de acceso indirecto (DY).

(2) Control de datos de dispositivo

Los controles de datos de dispositivo para el dispositivo usado mediante instrucciones básicas e instrucciones de aplicación en módulos CPU son como los indicados seguidamente:

(a) Cuando se están usado datos BIN:

• No se devuelve error incluso si el resultado de la operación se encuentra en desbordamiento.
El flag de acarreo tampoco continua activado en dichas condiciones.

(b) Cuando se usan datos BCD

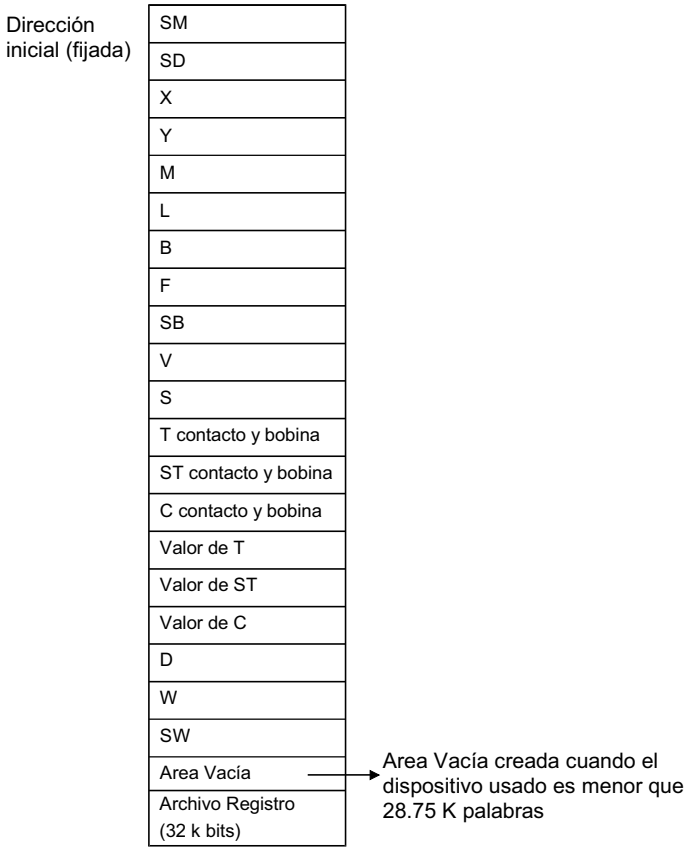
1) Cada dígito se controla mediante valor BCD (0 a 9).
Se retorna un error si los dígitos individuales están fuera del rango que va de 0 a 9 en hexadecimal, es decir de A a F.

2)No se devuelve error incluso sí los resultados de la operación overflow o underflow. En ningún caso no se activa el acarreo.

(c) Cuando se usan datos decimales en coma flotante
Se retornan errores en los siguientes casos:
• Cuando el valor en coma flotante es 0
• Cuando el valor absoluto del dato decimal en coma flotante es 1.0×2^{-127} o inferior
• Cuando el valor absoluto del dato decimal en coma flotante es 1.0×2^{128} o superior

(d) Cuando se usan caracteres de cadena de caracteres
No se prescribe ningún control de datos.

(3) Si la asignación interna de dispositivo del usuario es cambiada mediante parámetros de asignación de dispositivo, tales asignaciones de dispositivo son hechas en el orden siguiente: Si la asignación del dispositivo usado es menor que 28.75 k palabras, el área posterior al dispositivo usado estará vacía.



Comentarios

1) Referirse al manual de usuario (Explicación de Funciones, Programación fundamental) del módulo de CPU usado o el manual de programación de QnACPU (Básico), para como cambiar la asignación interna de dispositivos de usuario.

3.7 Condiciones de ejecución de instrucciones.

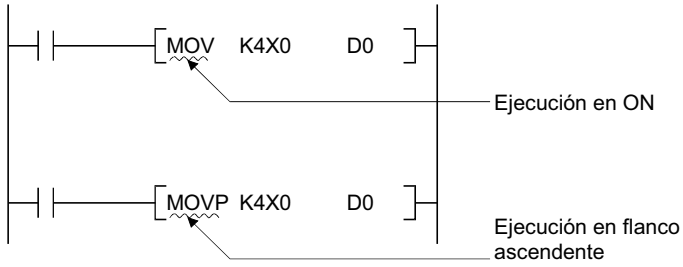
Existen los siguientes cuatro tipos de condiciones de ejecución existen para la ejecución de las instrucciones de secuencia, instrucciones básicas e instrucciones de aplicación del módulo de CPU:

- Ejecución no condicional ..... Instrucciones ejecutadas sin la consideración del estado de ON/OFF del dispositivo.
  Ejemplo: LD X0, OUT Y10
- Ejecutado a ON ..... Instrucciones ejecutadas mientras la condición de entrada esta en estado ON
  Ejemplo: instrucción MOV, instrucción FROM
- Ejecutado en flanco ascendente..... Instrucciones ejecutadas solamente en flanco ascendente de la condición de entrada (cuando va de OFF a ON)
  Ejemplo: instrucción PLS, instrucción MOVP
- Ejecutado en flanco descendente ..... Instrucciones ejecutadas solamente en flanco descendente de la condición de entrada (cuando va de ON a OFF)
  Ejemplo: instrucción PLF

Para bobina o instrucción básica equivalente o instrucciones de aplicación, dónde la misma instrucción puede ser designada bien para ejecuciones en ON o ejecuciones de flanco ascendente, una "P" se añade después del nombre de la instrucción para especificar la condición de ejecución.

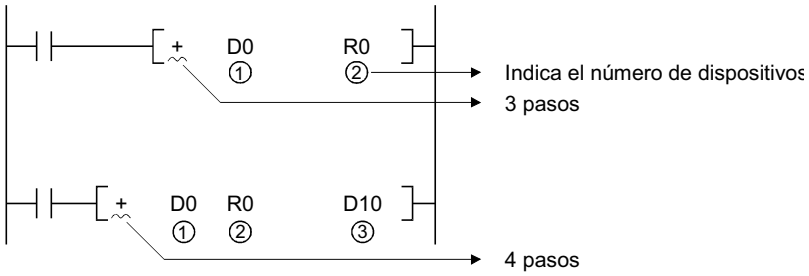
- Instrucciones para ser ejecutadas en ON — Nombre de la instrucción
- Instrucciones para ser ejecutadas en flanco ascendente — Nombre de la instrucción + P

Ejecución en ON y ejecución en flanco ascendente para la instrucción MOV son designadas como se muestra:



3.8 Contando el número de pasos

El número de pasos en secuencia de instrucciones, instrucciones básicas e instrucciones de aplicación de módulo CPU difiere dependiendo de sí es posible o no el ajuste indirecto de dispositivos.
El número básico de pasos para instrucciones elementales e instrucciones de aplicaciones es calculado mediante suma del número de dispositivo y 1.
Por ejemplo, la instrucción "+ " sería calculada como se muestra:



(1) Condiciones para incrementar el número de pasos
El número de pasos es incrementado sobre del número de pasos básicos en el caso dónde un dispositivo es usado, el cual es designado indirectamente o para el cual el número de pasos es incrementado.
(a) Cuando el dispositivo es designado indirectamente
En casos dónde la designación indirecta se realiza mediante @, el número de pasos es incrementado 1 paso sobre del número de pasos básico.
Por ejemplo, cuando una instrucción MOV de 3 pasos es designada indirectamente (por ejemplo: MOV K4X0 @D0), un paso es añadido y la instrucción pasa a ser de 4 pasos.

(b) Dispositivos donde el número de pasos se incrementan

Dispositivos donde el número de pasos se incrementan	Pasos incrementados	Ejemplo
Módulos de función inteligente dispositivo/módulo de función especial	1	MOV U4\G10 D0
Dispositivo directo de conexionado		MOV J3\B20 D0
Número de serie de acceso a registros de archivo		MOV ZR123 D0
Constantes de 32 bits		DMOV K123 D0
Constante Número Real		EMOV E0.1 D0
Constantes de cadenas de caracteres	Para números pares: (número de caracteres)/2 Para números impares: (número de caracteres + 1)/2	$MOV "123" D0

(c) En los casos donde las condiciones descritas (a) y (b) solapados arriba, el número de pasos pasa a ser la culminación de los dos.
Por ejemplo, si MOV U1\G10 ZR123 ha sido designado, 1 paso es añadido por el registro de buffer designado y 1 paso es añadido por el acceso de número de serie de la designación de archivo del registro, haciendo un total de 2 pasos sumados.

3.9 Funcionamiento cuando las instrucciones OUT, SET/RST, o PLS/PLF utilizan el mismo dispositivo

Seguidamente se describe el funcionamiento para la ejecución de instrucciones múltiples de OUT, SET/RST, o PLS/PLF que usan el mismo dispositivo en un ciclo de scan.

(1) Instrucciones de salida (OUT) que usan el mismo dispositivo

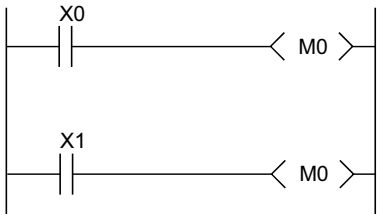
No programar más de una instrucción OUT usando el mismo dispositivo en un ciclo de scan.

Si las instrucciones OUT usando el mismo dispositivo son programadas en un ciclo de scan, el dispositivo especificado pasará de ON o OFF cada vez que la instrucción OUT es ejecutada, dependiendo del resultado de la operación del programa hasta la instrucción válida OUT.

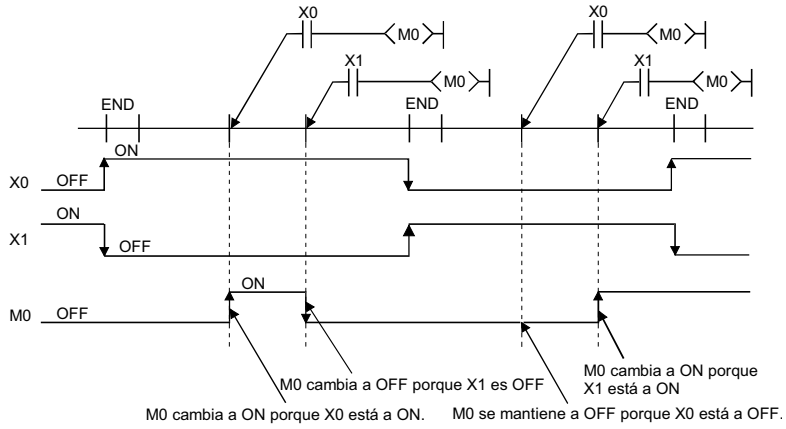
Como activando a ON o OFF el dispositivo es determinado cuando cada instrucción OUT es ejecutada, el dispositivo podría cambiar de ON y OFF repetidamente durante un ciclo de scan.

El siguiente diagrama muestra un ejemplo del circuito que conmuta el mismo relé interno (M0) con las entradas X0 y X1 ON y OFF.

[Circuito]



[Diagrama de Tiempos]



Con el refresco de la CPU, cuando la salida (Y) se especifica por la instrucción de salida, el estado ON/OFF de la última instrucción del ciclo de scan será emitida.

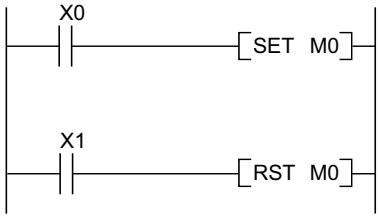
(2) Instrucciones SET/RST usando el mismo dispositivo

(a) La instrucción de SET cambia a ON el dispositivo especificado cuando el comando SET está en ON y no hace nada cuando el comando SET está en OFF.
Por esta razón, cuando dos o más instrucciones de SET usan el mismo dispositivo en un ciclo de scan, el dispositivo especificado estará en ON si cualquiera de los comandos de SET está en ON.

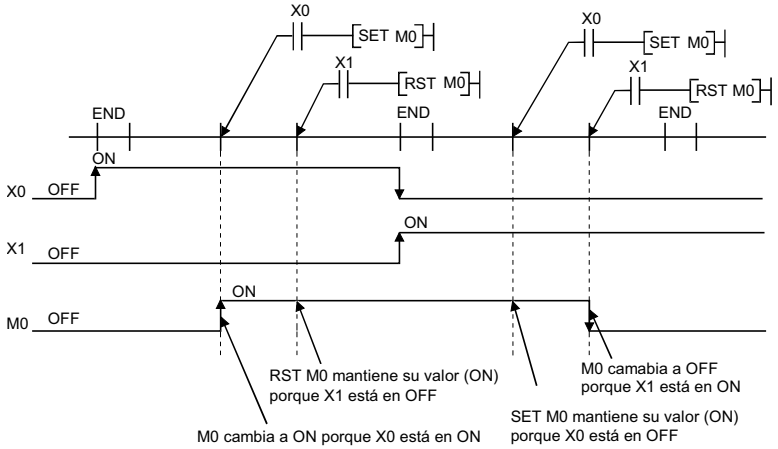
(b) La instrucción de RST pone a OFF el dispositivo especificado cuando el comando de RST está en ON y no hace nada cuando el comando de RST está en OFF.
Por esta razón, cuando dos o más instrucciones de RST usan el mismo dispositivo en un ciclo de scan, el dispositivo especificado estará en OFF si cualquiera de los comandos de RST está en ON.

(c) Cuando la instrucción de SET y la instrucción de RST usando el mismo dispositivo están programadas en un ciclo de scan, la instrucción de SET pone en ON el dispositivo especificado cuando el comando de SET está en ON y la instrucción de RST pone en OFF el dispositivo especificado cuando el comando de RST se encuentra en ON.
Cuando los comandos de SET y RST se encuentran en modo OFF, el estado ON/OFF el estado del dispositivo especificado no cambiará.

[Circuito]



[Diagrama de Tiempos]


X0
SET M0
X1
RST M0
END
ON
OFF
M0
RST M0 mantiene su valor (ON) porque X1 está en OFF
M0 cambia a ON porque X0 está en ON
M0 camabia a OFF porque X1 está en ON
SET M0 mantiene su valor (ON) porque X0 está en OFF

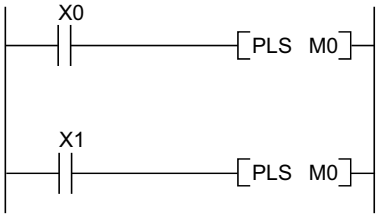
(3) Instrucciones PLS usando el mismo dispositivo

La instrucción PLS pone a ON el dispositivo especificado cuando el comando PLS cambia a ON estando previamente en OFF. Pone a OFF el dispositivo especificado en cualquier otro estado (OFF → OFF, ON → ON, y ON → OFF).

Cuando dos o más instrucciones PLS usando el mismo dispositivo son programadas en un ciclo de scan, cada instrucción de PLS cambia a ON el dispositivo especificado cuando el comando PLS correspondiente cambia a ON estando previamente a OFF y cambia a OFF el dispositivo especificado en cualquier otro estado.

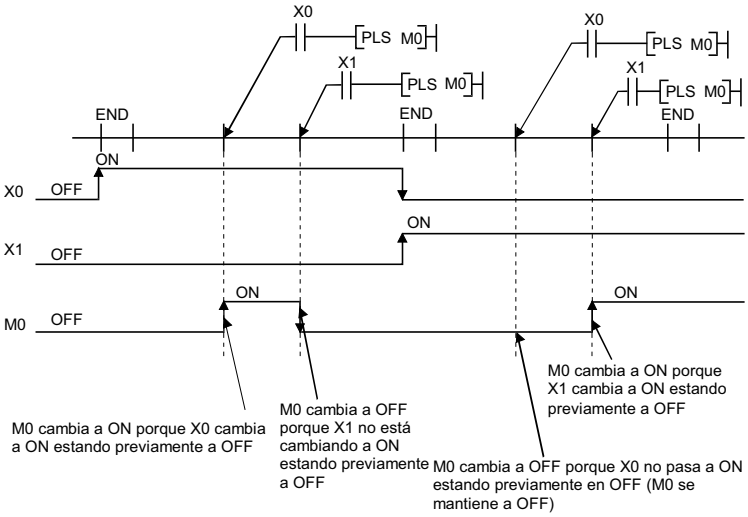
Por esta razón, cuando dos o más instrucciones PLS usan el mismo dispositivo son programadas en un ciclo de scan, el dispositivo que ha sido cambiado a ON por el comando PLS puede no cambiar a ON otra vez en todo el ciclo de scan.

[Circuito]

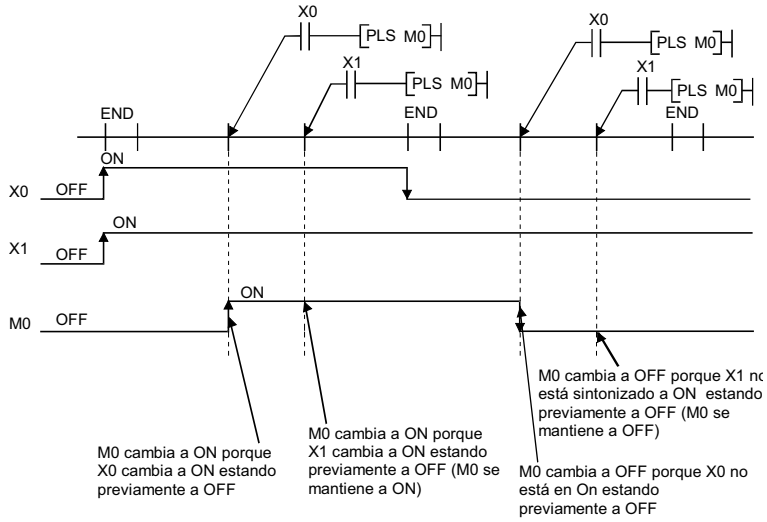


[Diagrama de Tiempos]

• El periodo ON/OFF del contacto X0 y X1 es diferente. (El dispositivo especificado no cambiará a ON en todo el ciclo de scan.)


X0
PLS M0
X1
PLS M0
END
ON
OFF
M0 cambia a ON porque X0 cambia a ON estando previamente a OFF
M0 cambia a OFF porque X1 no está cambiando a ON estando previamente a OFF
M0 cambia a OFF porque X0 no pasa a ON estando previamente en OFF (M0 se mantiene a OFF)
M0 cambia a ON porque X1 cambia a ON estando previamente a OFF

• Los contactos X0 y X1 cambiarán a ON estando previamente a OFF al mismo tiempo.



3 - 29

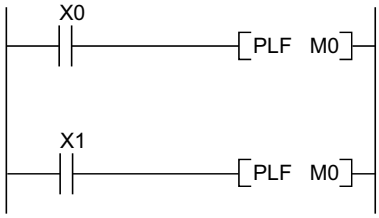
(4) Instrucciones PLF usando el mismo dispositivo

La instrucción PLF cambia a ON el dispositivo especificado cuando el comando PLF cambia a ON estando previamente a OFF. Pone a OFF el dispositivo especificado estando en cualquier otro estado (OFF → OFF, OFF → ON, y ON → ON).

Cuando dos o más instrucciones PLF usando el mismo dispositivo son programadas en un ciclo de scan, cada instrucción PLF pone a estado OFF el dispositivo especificado cuando el comando PLF correspondiente cambia a estado OFF estando previamente a ON y cambia a estado OFF el dispositivo especificado en cualquier otro caso.

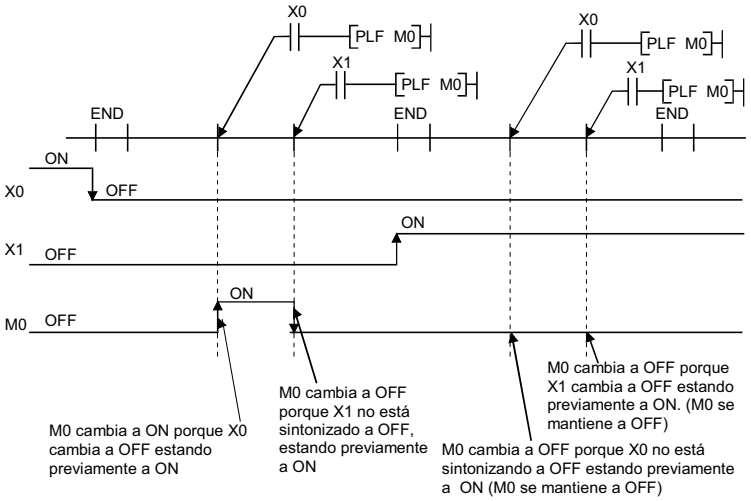
Por esta razón, cuando dos o más instrucciones PLF usando el mismo dispositivo son programadas en un ciclo de scan, el dispositivo que ha sido cambiado a estado ON por el comando PLF puede no cambiar a ON otra vez en todo el ciclo de scan.

[Circuito]

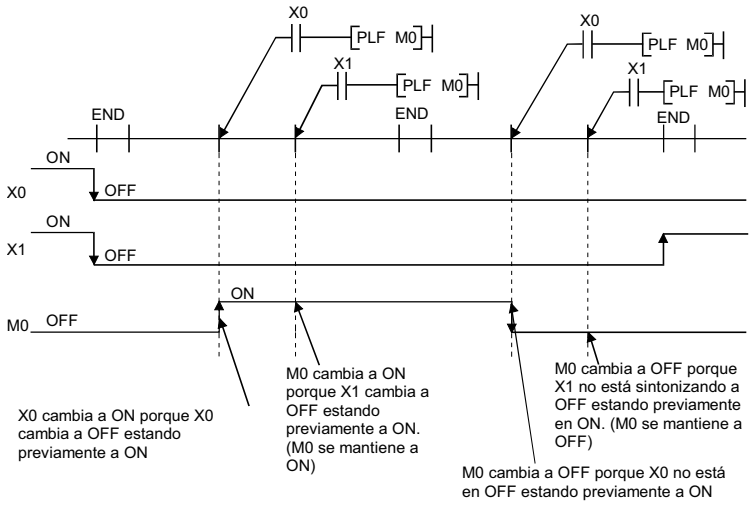


[Diagrama de Tiempos]

• El periodo ON/OFF de los contactos X0 y X1 es diferente. (El dispositivo especificado no cambia a ON en todo el ciclo de scan.)



• Los contactos X0 y X1 cambian a OFF estando previamente a ON en el mismo periodo.



3.10 Precauciones para el uso archivo de registros

Este apartado explica las precauciones para el uso de archivo de registros respecto los modelos QCPU y QnACPU.

(1) Módulos CPU que no pueden utilizar archivo de registros.
El modelo Q00JCPU no puede usar archivo de registros.
Cuando se usan archivos de registros, usar los módulos de CPU que no sean del modelo Q00JCPU.

(2) Configuración de los archivos de registros a ser usados
Cuando se usan archivos de registros, los archivos de registros a ser usados deben ser configurados con el “PLC parameters” o la instrucción QDRSET. (El “PLC parameters” del modelo Q00CPU y Q01CPU no necesitan ser configurados una vez están configurados en "Use file register".)
Si los archivos de registros a ser usados, no han sido configurados, una operación normal no puede realizarse con las instrucciones que utilizan los archivos de registros.

Observaciones

Si los archivos de registro no han sido configurados en el “PLC parameter”, puede ser creado un programa que usa los archivos de registros, y no se producirá error cuando el programa sea escrito y ejecutado en el módulo de CPU.
A destacar que la lectura/escritura del dato correcto to/from el registro de archivo no puede ser representado.

(3) Seguridad del área de registro
(a) El modelo de alto nivel QCPU, procesador CPU, QnACPU
Cuando se usan archivos de registros, almacenar los archivos de registro de la tarjeta memoria RAM estándar para asegurar el área archivos de registro.

(b) Modelo básico de QCPU (excepto Q00JCPU)
El área de archivos de registro ha sido guardada en la RAM de antemano.
El usuario no necesita guardar el área del archivo de registro.

La siguiente tabla indica las memorias que pueden usar archivos de registro en cada módulo CPU.

Memoria	QnACPU	Modelo de alto nivel QCPU, Procesamiento CPU	Modelo básico QCPU (excepto Q00JCPU)
Standard RAM	×	○	○
Tarjeta de memoria ∗1,∗2	○	○	×

○ : Puede ser registrado × : Puede no ser registrado.

∗1: Cuando es usada la memoria flash, solo leer archivos de los registros que pueden ser representados. (No se puede escribir la flash ROM.)
∗2: Cuando se usa la E^2PROM, escribir en la E^2PROM puede ser representado con la instrucción PROMWR.

Comentarios

Para el método de configuración de archivos de registros y el área del método de registro de seguridad, referirse al manual de usuario (explicación de funciones, programa fundamental) de los módulos usados de CPU o del manual de programación QnACPU (Fundamental).

(4) Designación del número de archivos de registro superior al número registrado de puntos

No ocurrirá un error si el dato es escrito o leido un (to/from) los registros de archivo que tienen números mayores que el número de puntos registrados.

Sin embargo, notar que en la lectura/escritura correcta de datos (to/from) los registros de archivo no pueden ser representados.

(5) Registros de archivo especificando método

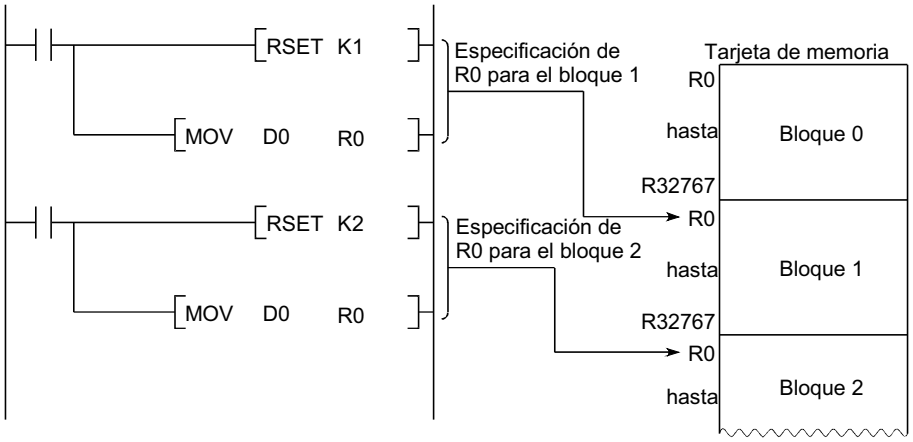
Hay un método de conmutación de bloques y un método de acceso numérico para especificar los registros de archivo.

(a) Método de conmutación de bloques.

En el método de conmutación de bloques, especificar el número de archivos de registro usados en unidades de 32k bits (un bloque).

Para archivos de registro de 32k bits o superior, especificar los archivos de registros por conmutación del número de bloque a ser usado con la instrucción RSET.

Especificar cada bloque de R0 hasta R32767.

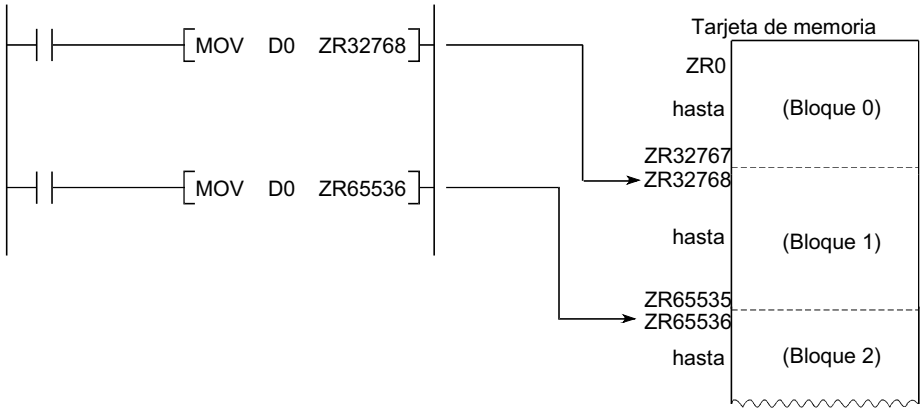


(b) Método de acceso a número de serie

En el método de acceso a número de serie, especificar los archivos de registros posteriores a 32k bits con números consecutivos de dispositivo.

Los archivos de registro de bloques múltiples pueden ser usados como archivos de registro consecutivos.

Usar "ZR" como el nombre de dispositivo.

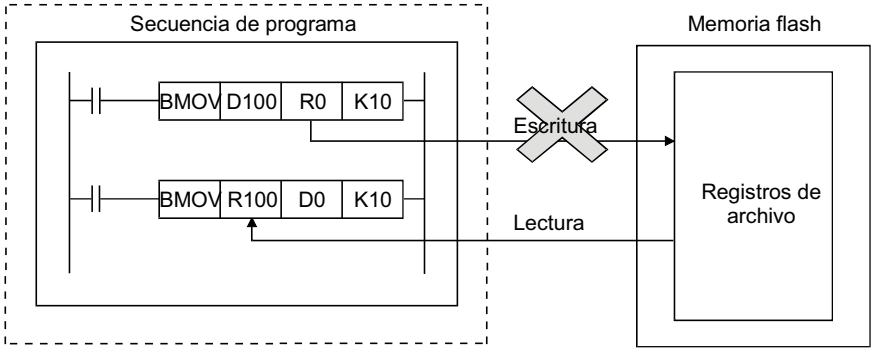

MOV D0 ZR32768
MOV D0 ZR65536
Tarjeta de memoria
ZR0
hasta
(Bloque 0)
ZR32767
ZR32768
hasta
(Bloque 1)
ZR65535
ZR65536
hasta
(Bloque 2)

3.11 Precauciones para el uso de memoria RAM

En esta sección se explican las precauciones para el uso de memoria flash respecto los modelos de QCPU y QnACPU.

(1) Las siguientes memorias flash pueden ser usadas con los modelos QCPU o QnACPU.
QCPU: Tarjeta flash
QnACPU: SRAM + tarjeta de memoria del tipo memoria flash.

(2) La memoria flash permite solo leer en una secuencia de programa.
(Escribir en la memoria flash puede no puede hacerse en un programa.)



Cuando se usa memoria flash para los registros de archivo, escribir datos con anterioridad.

(a) QCPU
Usando Developer, escribir datos en la tarjeta flash.

(b) QnACPU
Usando una lectura / escritura de una "IC memory card", escribir datos en la SDRAM + tarjeta de memoria de tipo memoria flash.

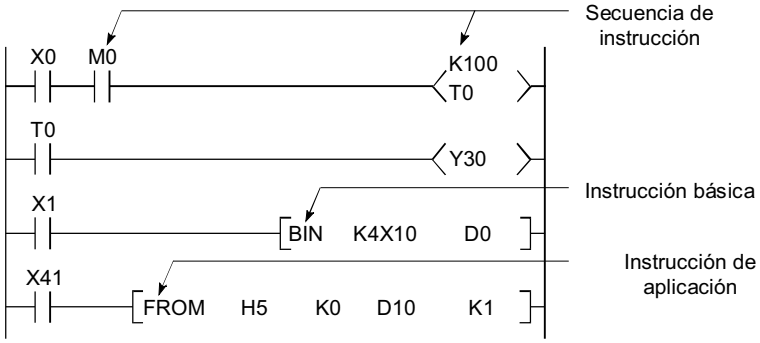
4 CONFIGURACIÓN DE LA SECUENCIA DE PROGRAMA Y CONDICIONES DE EJECUCIÓN

La secuencia de programa y los programas de tipo SFC pueden ser ejecutados con el modelo QCPU de altas prestaciones.
Este capítulo describe la configuración de la secuencia de programa y ejecución de condiciones.
Los programas SFC no se describen en este manual.
Para detalles de programas SFC, referirse al modelo QCPU (modo Q)/QnACPU Programación del Manual (SFC).

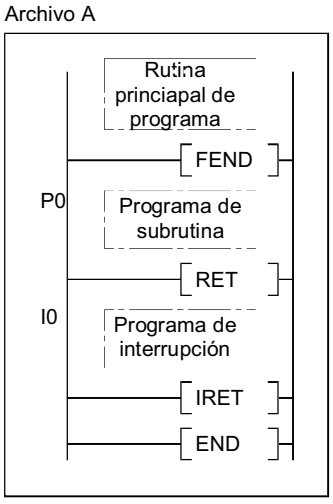
4.1 Secuencia de programa

(1) Definición de secuencia de programa

(a) Una secuencia de programa consiste en una secuencia de instrucciones, instrucciones básicas e instrucciones de aplicación, etc.



(b) Hay 3 tipos de secuencia de programa: programas de rutina principal, subrutina de programa y programas de interrupción.
Para detalles acerca de estos programas, ver las siguientes secciones de este manual.:
- Programas de rutina principal: Sección 4.1.1
- Programas de subrutina: Sección 4.1.2
- Programas de interrupción : Sección 4.1.3



Comentarios

Para detalles de la secuencia de instrucciones, instrucciones básicas e instrucciones de aplicación, referirse al modelo " QCPU (Modo Q)/QnACPU Manual de programación (Instrucciones comunes)".

(2) Secuencia de programa escribiendo formato

La programación para secuencias de programa se habilita usando los modos ladder y lista de instrucciones.

(a) Modo Ladder

- El modo ladder está basado en el control de relés de la secuencia ladder. La programación de expresiones es similar a la del control de relés de secuencia ladder.
- La programación de lenguaje simbólico de contactos se usa en los bloques de unidades ladder.
  El bloque ladder es la unidad más pequeña de operación secuencia de programa, con el principio en la parte izquierda y el final en la parte derecha del bus de contactos.

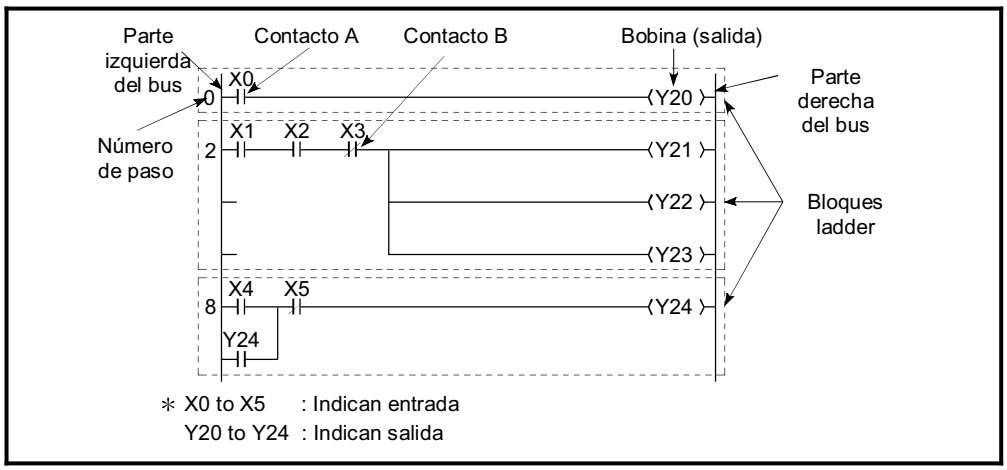


Fig.4.1 Bloque Ladder

(b) Modo de listado de instrucciones.

El modo de listado de instrucciones usa instrucciones dedicadas en vez de simbología de contactos, simbología de bobinas, etc., usados en el modo ladder.

Contacto a, contacto b e instrucciones de bobina son como las siguientes:

- Contacto A.....LD, AND, OR
- Contacto B.....LDI, ANI, ORI
- Bobina.....OUT

(3) Operación de programa

En secuencia de programa, las instrucciones son ejecutadas por orden empezando desde el paso 0 y acabando en la instrucción END/FEND.
Con el modo ladder, las instrucciones en el bloque ladder son ejecutadas por orden empezando desde la parte izquierda del bus hasta la parte derecha del mismo. Cuando un bloque ladder es completado, el bloque ladder de la línea siguiente será ejecutado.

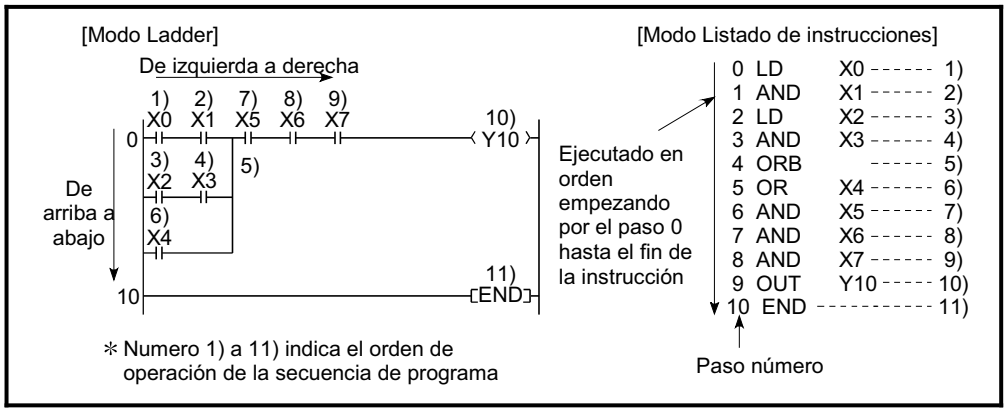


Fig.4.2 Secuencia de funcionamiento de programa

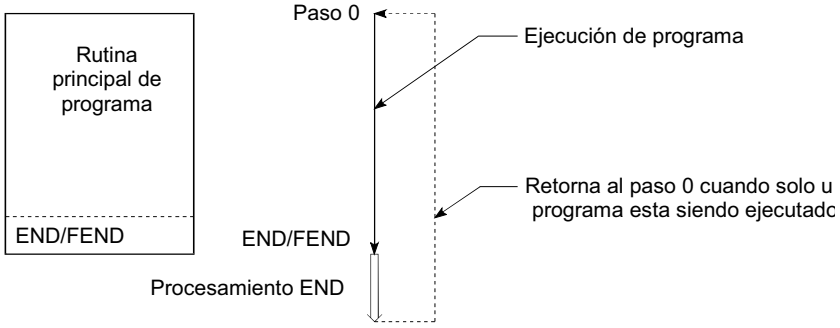
4.1.1 Programas de rutina principal.

(1) Definición de programa de rutina principal

(a) Una rutina principal de programa empieza desde el paso 0 y acaba en la instrucción END/FEND. *1

(b) En el programa principal de rutina, las instrucciones son ejecutadas desde el paso 0 hasta la instrucción END/FEND.

1) Si solo un programa se ejecuta, el programa se ejecuta desde el paso 0 después de que la instrucción END/FEND sea ejecutada y la instrucción END sea procesada.



2) Si múltiples programas están siendo ejecutados, cada uno de ellos será ejecutado de acuerdo con la condición de ejecución de designación después de que la instrucción END/FEND sea ejecutada.

(2) Condiciones de ejecución para programas de rutina principal *2

Si están siendo ejecutados múltiples programas, los siguientes cinco tipos de condiciones de ejecución pueden ser designados por el programa en los parámetros del PLC de acuerdo con la aplicación.

- Programa de ejecución inicial :Ver Sección 4.2.1.
- Programa de ejecución por SCAN :Ver Sección 4.2.2.
- Programa de ejecución de baja Velocidad :Ver Sección 4.2.3.
- Programa de mantenimiento :Ver Sección 4.2.4.
- Programa de ejecución de SCAN fijo :Ver Sección 4.2.5.

Comentarios

∗1: Para detalles referentes a la instrucción END/FEND, referirse a "QCPU (Q modo/QnACPU Manual de Programación (Instrucciones comunes)".

∗2: Si solo un programa es ejecutado, se procesa bajo la condición de “Programa de ejecución tipo SCAN” sin designación mediante el programa en los parámetros del PLC.

4.1.2 Programas de subrutina

(1) Definición de programa de subrutina

(a) Un programa de subrutina empieza con el puntero (P) y acaba con la instrucción RET.

(b) Una subrutina de programa es ejecutada solamente cuando se llama mediante la instrucción CALL (por ejemplo CALL(P), FCALL(P)) desde el programa de rutina principal.

(c) Aplicación de programa de subrutina.

1) El número de pasos de programa puede ser reducido mediante el uso de una subrutina de programa como un programa el cual es ejecutado varias veces en un ciclo de scan.

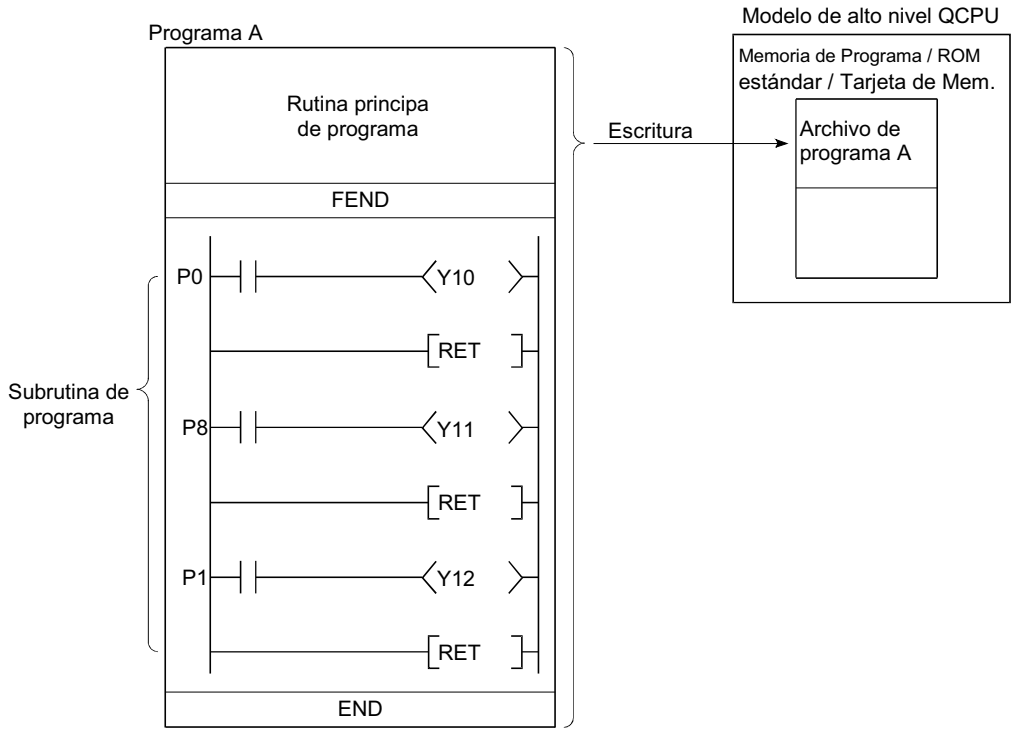
2) El número de pasos de un programa ejecutado constantemente puede ser reducido mediante el uso de una subrutina de programa como un programa el cual es ejecutado solamente cuando una condición dada es satisfecha.

(2) Gestión de programas de subrutina

Los programas de subrutinas son creados después del programa de rutina principal (después de la instrucción FEND), y la combinación del programa principal y la subrutina puede ser gestionado como un solo programa.

(a) Cuándo es creado después del programa principal de rutina.

- Un programa de subrutina es creado entre la rutina principal de programa FEND y las instrucciones END.
- Como no hay restricciones respecto el orden en el cual los programas de subrutina son creados, no es necesario poner los punteros en orden ascendente cuando se están creando programas de subrutina múltiples.
- Punteros locales o punteros comunes pueden ser usados. ∗



Comentarios

∗: Ver sección 10.9 para detalles referentes a punteros locales y comunes. Ver sección 10.8 para detalles a cerca de anidamiento de programas de subrutina.

(b) Usando programas de subrutinas como programas separados.
Los programas de subrutina pueden también ser utilizados por separado, programas individualizados (programas en estado de stand-by). (Ver sección 4.2.4 para detalles referentes al modo stand-by).

4.1.3 Programas de interrupción

(1) Definición de programa de interrupción

(a) Un programa de interrupción empieza desde el puntero de interrupción (I□), y acaba en la instrucción IRET. ∗1

(b) Los programas de interrupción son ejecutados solamente cuando ocurre un factor de interrupción. ∗1

Observaciones

Los punteros de interrupción incluyen un puntero designado sólo para la función de interrupción de alta velocidad (I49).
Cuando se ha usado I49, no usar programas de interrupción, los cuales usen el orden de interrupción de los punteros I0 a I48, I50 a I255, y programas de tipo de ejecución tipo scan fijo.
Si cualquier programa de ejecución del tipo scan fijo o similar está activado, el programa de interrupción que use I49 no puede ser ejecutado en intervalos

preajustados de ciclos de interrupción.
Ver sección 7.20 para más detalles a cerca de las funciones de interrupción de alta velocidad.

Comentarios

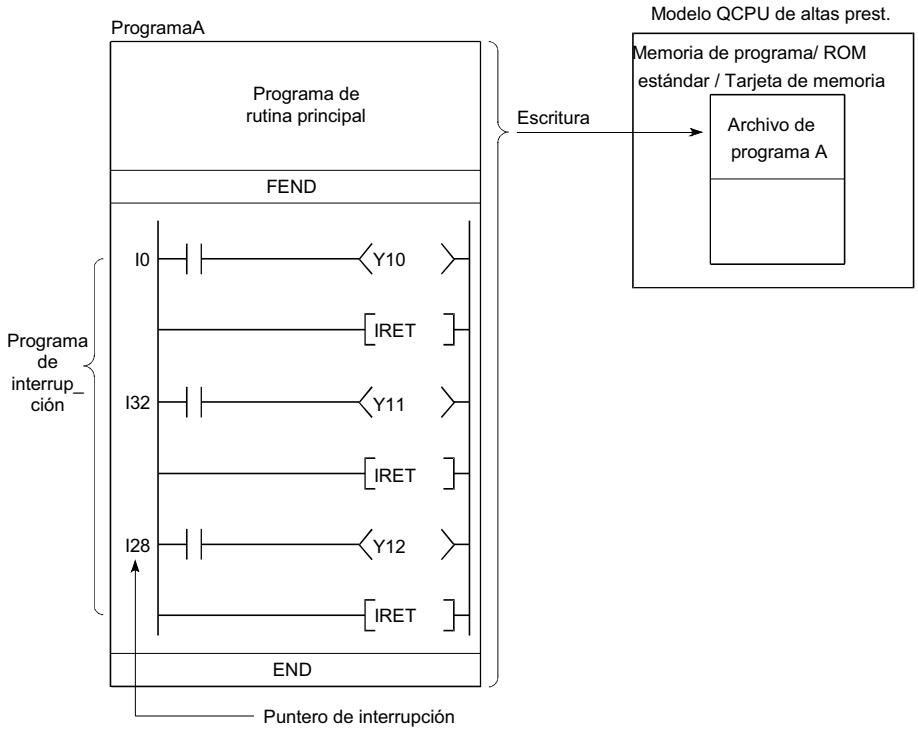
∗1: Ver sección 10.10 para más detalles acerca de factores de interrupción y punteros de interrupción.

(2) Gestión de programas de interrupción.

Los programas de interrupción están creados después del programa de rutina principal (después de la instrucción FEND). Por lo tanto, la combinación del programa principal y el programa de subrutina puede ser tratado como un solo programa.

(a) Cuando se han creado después de la rutina principal de programa.

- Se crea un programa de interrupción entre las instrucciones FEND y END del programa de rutina principal.
- Como no hay restricciones en el orden en el cual son creadas las interrupciones del programa, no es necesario habilitar los punteros de interrupción en sentido ascendente cuando se están creando programas de interrupciones múltiples.



(b) Usando programas de interrupción como programas individualizados.

Los programas de interrupción pueden también ser ejecutados por separado como programas discretos (programas en modo stand-by). (Ver sección 4.2.4 para más detalles en programas de tipo stand-by).
Sin embargo, el mismo número de punteros de interrupción de programa de I32 a I47, las instrucciones IMASK e EI son requeridas para obtener permiso para la interrupción. ∗1

(3) Ejecutando programas de interrupción.

(a) Para poder ejecutar una interrupción de un programa con los punteros de interrupción I32 a I47, las instrucciones IMASK y EI son requeridas para obtener permiso para la interrupción. ∗1

1) Si un factor de interrupción ocurre antes de que haya el estado que permita la interrupción, el programa de interrupción correspondiente al factor será ejecutado cuando el estado del led de “interrupción permitida se establezca.

2) Si un factor de interrupción ocurre durante un estado de STOP/PAUSE, la interrupción del programa correspondiente con dicho factor será ejecutado cuando el estado de "interrupción permitida" establecida retornando al estado de RUN.

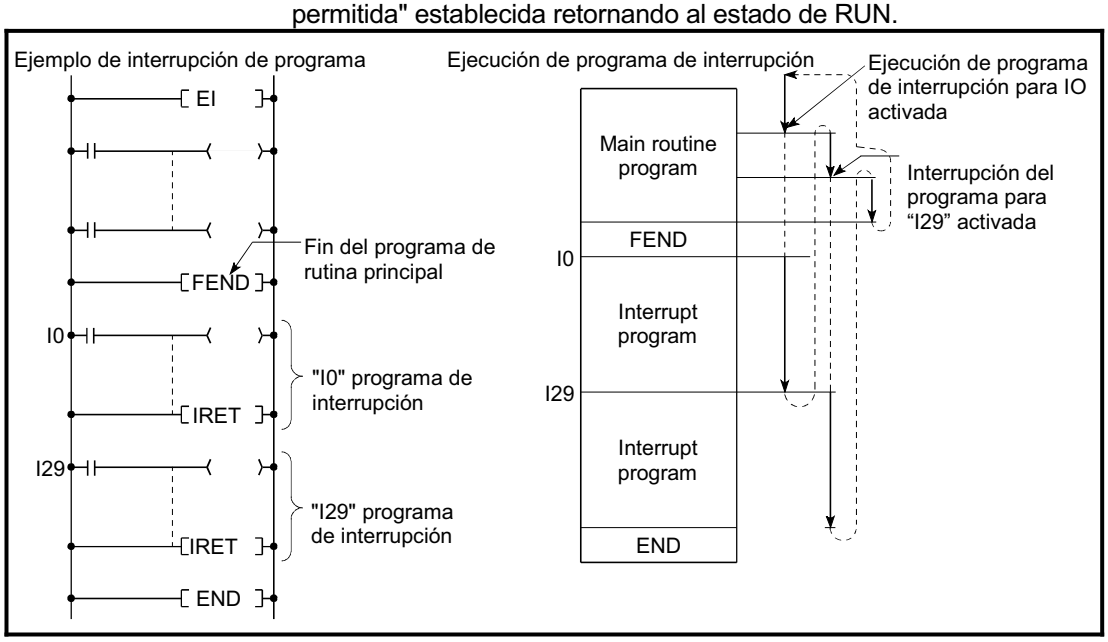


Fig.4.3 Ejecución de programa de interrupción

(b) Cuando ocurre un factor de interrupción, el programa de interrupción con el número de puntero de interrupción correspondiente a ese factor es ejecuta. Pero, la ejecución del programa de interrupción varia de acuerdo con la condición en ese momento.

1) Si dos o más interrupciones ocurren al mismo tiempo:
Los programas de interrupción son ejecutados, empezando con el correspondiente al numero de puntero de interrupción (I ⬚) de la prioridad más alta. ∗2
Los restantes programas de interrupción se mantienen en estado de stand-by hasta que el procesado de la interrupción de programa se completa.
Si el mismo factor de interrupción ocurre, como el que está siendo ejecutado antes de que la interrupción de programa sea ejecutada, el factor de interrupción será almacenado en memoria. Después de que el programa de interrupción haya sido procesado, el mismo programa de interrupción será ejecutado otra vez.

2) Cuando una instrucción está siendo ejecutada:
El programa de interrupción puede ser ejecutado mediante la ejecución de una interrupción de una instrucción en el programa de rutina principal. Cuando el mismo dispositivo es usado en el programa de rutina principal y el programa de interrupción, el dato del dispositivo puede ser separado.
Para prevenir la separación del dato de dispositivo, se deben acotar las siguientes medidas.
(a) No especificar el dispositivo, para el dato del cual será escrito en el programa de interrupción, directamente en el programa de rutina principal, sino usa otros dispositivos y realice un intercambio de datos con instrucciones de trasferencia, etc.

(b) Si aparece algún inconveniente cuando la instrucción es interrumpida en el programa de rutina principal, se debe ejecutar después de deshabilitar la interrupción con la instrucción DI.

No obstante, como la interrupción de programa no interrumpirá durante el acceso al dispositivo de cada argumento de la instrucción, no ocurrirá separación en base al argumento.

3) Interrupción durante el refresco de la red:
Si un factor de interrupción ocurre durante una operación de refresco de la red, la operación de refresco de red es suspendida y la interrupción del programa se ejecuta.
Esto significa que "la garantía de bloques en datos cíclicos en cada estación" no puede ser asegurada mediante el uso de dispositivos designados como un destino de operación de refresco de red MELSECNET/H. ∗3

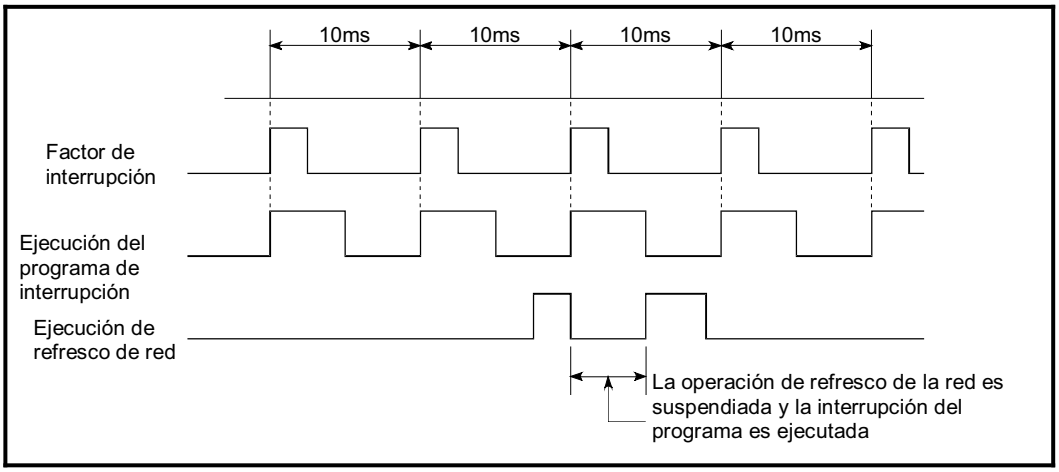

10ms
10ms
10ms
10ms
Factor de interrupción
Ejecución del programa de interrupción
Ejecución de refresco de red
La operación de refresco de la red es suspendiada y la interrupción del programa es ejecutada

Fig.4.4 Operación de refresco de red durante la interrupción.

4) Interrupción durante el proceso END:
Si un factor de interrupción ocurre durante el periodo de espera de END cuando se reestablece un ciclo de scan constante la interrupción del programa correspondiente al factor será ejecutada.

Comentarios

∗1: Para detalles acerca de las instrucciones IMASK y EI, referirse al manual de programación "QCPU (modo Q)/QnACPU (Instrucciones comunes).
Para ejecutar programas entre I0 y I31 y entre I48 y I255, usar la instrucción EI para entrar los programas de interrupción en un estado de habilitación de interrupción.
∗2: Ver sección 10.10 para detalles sobre el rango de prioridad de programas de interrupción.
∗3: Para garantizar los bloques de unidad de estación en datos cíclicos, ver el manual de referencia de sistema de red "MELECNET/H."

(c) Cuando el programa de interrupción es ejecutado por defecto en el modelo QCPU de altas prestaciones, almacenado y la recuperación del valor del registro y el almacenado y recuperación del número de bloque de archivo de registro se efectúan en el tiempo de conmutado entre el programa de rutina principal y el programa de interrupción.
Referirse a la sección 10.6.2 para más detalles.

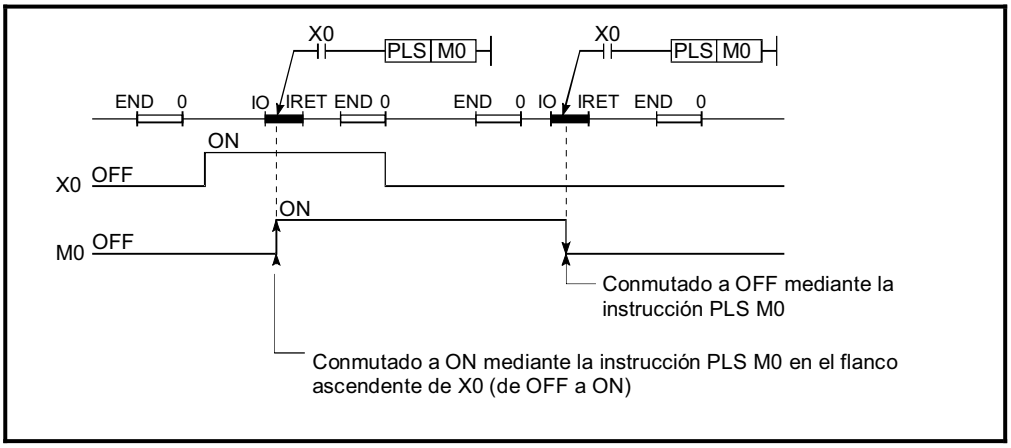
(4) Ejecución de alta velocidad de un programa de interrupción y tiempo de añadido

Por defecto, el modelo QCPU de altas prestaciones representa los siguientes procesos cuando se ejecuta una interrupción de programa:
• Guardar y restaurar un registro índice. (Ver sección 10.6.2)
• Guardar y restaurar el nombre del archivo de un registro de archivo en uso.
Los procesos listados arriba no se representan si la "Ejecución a alta velocidad" es seleccionada en la pestaña del "PLC System" en la ventana de dialogo "(PLC) Parameter".
Esto hará posible acortar la duración de tiempo de añadido requerido para ejecutar un programa de interrupción.

TIPO DE CPU	TIEMPO AÑADIDO (μ s)	
	Ejecución de alta velocidad no está seleccionada	Ejecución de alta velocidad está seleccionada
Q02CPU	380	230
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	165	100

(5) Restricciones en la creación de programa

(a) Un dispositivo que es conmutado a ON mediante la instrucción PLS en una interrupción de programa permanecerá a ON hasta que la interrupción de programa se ejecute otra vez.



(b) El estado DI (interrupción prohibida) se establece durante la ejecución de un programa de interrupción.
No ejecutar las instrucciones EI/DI en el programa de interrupción.

(c) Los temporizadores no pueden ser usados en interrupciones de programa. Como los temporizadores se usan con instrucciones OUT T□ para actualizar valores consignados y contactos de conmutación ON y OFF, el uso de un temporizador en una interrupción de programa haría imposible un tiempo de contaje normal imposible.

(d) Los siguientes comandos no pueden ser usados en el programa de interrupción.
• COM
• ZCOM

(e) Cuando el programa de interrupción (programa de ejecución de scan fijo) es ejecutado en un tiempo como el tiempo de scan o el tiempo de

ejecución, los valores del programa de interrupción (programa de ejecución de scan fijo) son añadidos al tiempo de medida.
De este modo, si el programa de interrupción (programa de ejecución de scan fijo), el valor almacenado en los siguientes registros especiales y el valor del monitoreo del GX Developer será mayor que cuando el programa de interrupción (de ejecución de scan fijo) no se ejecuta.

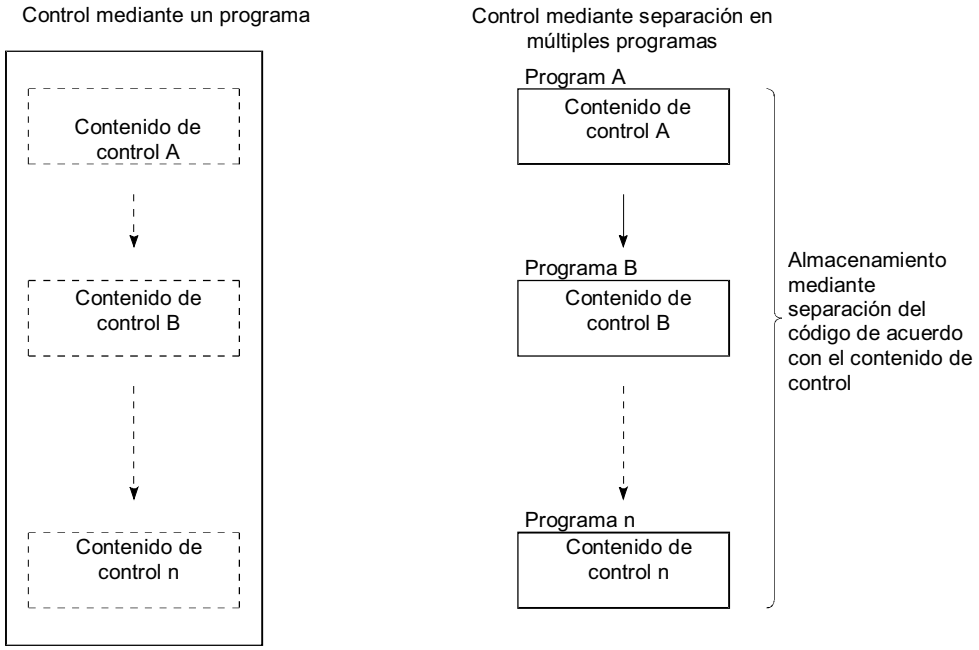
1) Registros especiales.
   - SD520, SD521: Tiempo de scan por defecto.
   - SD522, SD523: Tiempo de scan inicial.
   - SD524, SD525: Tiempo de scan mínimo.
   - SD526, SD527: Tiempo de scan máximo,
   - SD528, SD529: Tiempo de scan por defecto para baja velocidad.
   - SD532, SD533: Tiempo de scan mínimo para baja velocidad.
   - SD534, SD535: Tiempo de scan máximo para máxima velocidad.
   - SD540, SD541: Tiempo de procesado de END.
   - SD542, SD543: Tiempo de espera a scan constante.
   - SD544, SD545: Tiempo de ejecución acumulativa para ejecución de programas de baja velocidad
   - SD546, SD547: Tiempo de ejecución de para tipo de programas de baja velocidad.
   - SD548, SD549: Tiempo de ejecución de programa scan.
   - SD551, SD552: Tiempo de intervalo de servicio.

2) Valor de monitoreo de GX Developer
   - Medida de tiempo de Ejecución
   - Medida de tiempo de scan
   - Scan constante

4.2 Tipo de ejecución de programa

Los programas ejecutados por el modelo QCPU de altas prestaciones pueden ser almacenados en la memoria de programa,ROM estándar o tarjeta de memoria de los modelos QCPU de altas prestaciones.
Los programas pueden ser almacenados en la ROM estándar o tarjeta de memoria como un solo programa, pero también como programas múltiples mediante su división en programas separados para cada función de control.
Esto permite que el procedimiento de programación esté separado entre muchos diseñadores de programas, los cuales pueden diseñar programas separados para cada operación y pueden almacenarlos en la estándar ROM o la tarjeta de memoria.

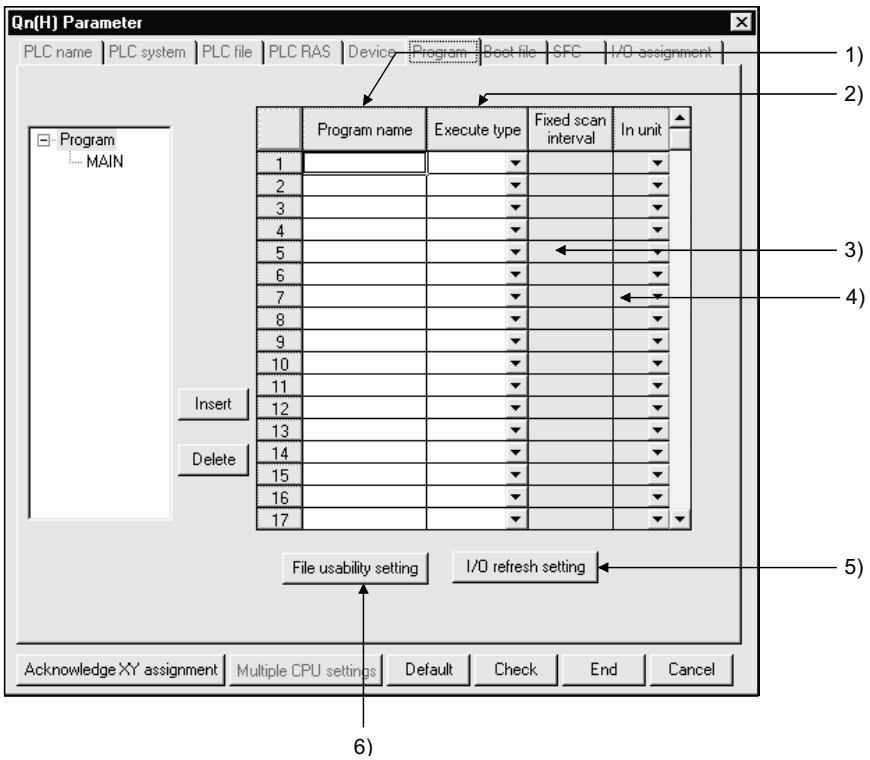
Cuando programas múltiples en un modelo QCPU de altas prestaciones, las configuraciones de programa “program name (file name)" y "execute type" de los programas deben ser designadas.


Control mediante un programa
Contenido de control A
Contenido de control B
Contenido de control n
Control mediante separación en múltiples programas
Program A
Contenido de control A
Programa B
Contenido de control B
Programa n
Contenido de control n
Almacenamiento mediante separación del código de acuerdo con el contenido de control

(1) Configuración del tipo de ejecución

(a) Para ejecutar varios programas, especificar un "Program name" y "Execute type" para cada programa en la pestaña "Program" en la ventana de diálogo “(PLC) Parameter".

El modelo QCPU de altas prestaciones ejecutará los programas seleccionados en el orden de configuración del tipo de ejecución especificada.



1) Nombre del programa
Esta columna es usada para especificar el nombre del programa (file 0name) del programa a ser ejecutado mediante el modelo QCPU de alto nivel.

2) Tipo de ejecución
Esta columna es usada para especificar el tipo de ejecución del programa definido en la columna "Program name". Ver sección (b).

3) Scan fijo
Esta columna es usada para especificar los intervalos de tiempo en que se ejecuta un programa.
La configuración del rango del scan fijo es determinada por las unidades de los intervalos de tiempo tal y como se muestra:
• En la unidad de "ms": 0.5 a 999.5
• En la unidad de "s": 1 a 60

4) En unidades
Esta columna especifica las unidades (ms/s) de los intervalos de scan fijos.

5) Configuración del uso de archivos
Configura si se usa el dato (archivo de registros de archivo, valor del dispositivo inicial, comentarios, dispositivos locales) configurados en la pestaña “PLC file” dentro de la ventana de diálogo “(PLC) Parameter” por programa. Los datos se configuran para cada programa. Por defecto, se selecciona la opción "Use PLC file setting". Si se selecciona la opción "Not used", la configuración del uso de archivo se realiza como se muestra seguidamente en la tabla.

Parámetros de configuración	Procesando cuando la opción “Not used" está seleccionada
Registro de archivo	Los registros de archivo no pueden ser usados en el programa.
Valor del dispositivo inicial	El valor del dispositivo inicial no se configura cuando el nombre del archivo de programa y el valor del dispositivo inicial es el mismo.
Comentario de archivo usado en un comando	Los comentarios no pueden ser usados en el programa.
Dispositivo local	Los dispositivos locales no son ocultados o restaurados al tiempo de la conversión del programa.

6) Configuración de refresco de E/S
El modelo QCPU de altas prestaciones, usa la configuración de refresco de E/S para actualizar entradas y salidas desde unos módulos de E/S y un módulo de función inteligente.
El botón de configuración de refresco de E/S es usado para actualizar el rango de selección de programas. Realizar la configuración del refresco de E/S para un programa de tipo de ejecución scan si se quiere recibir una entrada (X) o producir una salida (Y) antes de la ejecución del tipo de programa de scan fijo.

(b) Hay los siguientes 5 tipos de ejecución:

1) Ejecución Incicial (Inicial)
Este programa es ejecutado al alimentar por primera vez cuando se conmuta de STOP-ON. (Ver sección 4.2.1)

2) Ejecución scan (Scan)
Este tipo de programa es ejecutado una vez por ciclo de scan, empezando desde el scan que sigue la ejecución del programa de ejecución inicial. (Ver sección 4.2.2)

3) Ejecución de baja velocidad (Baja velocidad).
Este tipo de programa es ejecutado solamente cuando una configuración de scan constante se efectúa o cuando se configura un tiempo para la ejecución de programa de ejecución de baja velocidad.
- Cuando se realiza una configuración de scan constante, el programa es ejecutado durante el tiempo sobrante de programa de ejecución tipo scan.
- Cuando se configura un tiempo para la ejecución de un programa de tipo de ejecución de baja velocidad, el programa es ejecutado durante susodicho tiempo. (Ver sección 4.2.3.)

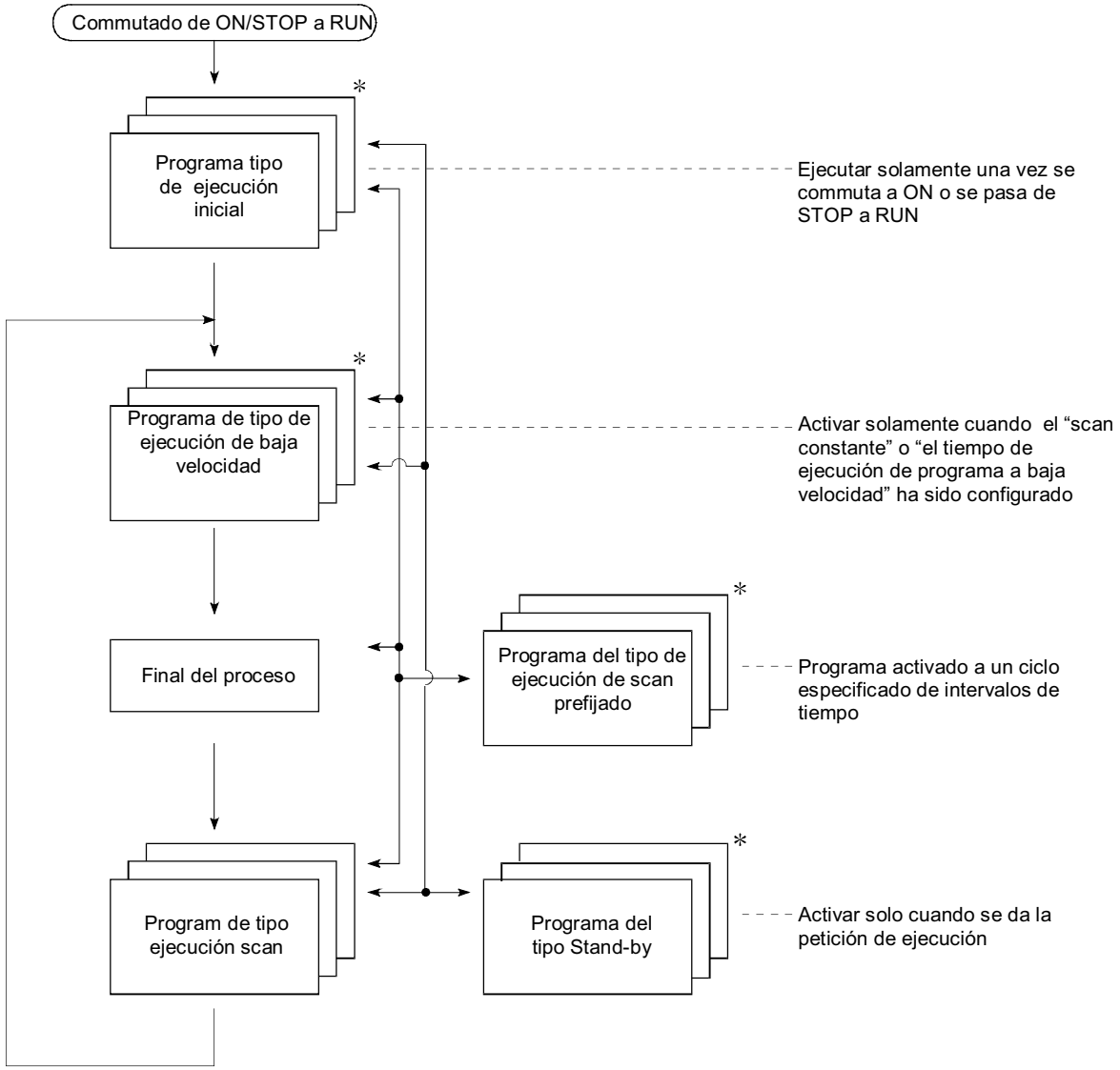
4) Stand-by (espera)
Este programa es ejecutado solamente cuando su ejecución es requerida. (Ver sección 4.2.4.)

5) Ejecución de scan fijo (Scan fijo)
Programa que es ejecutado en el tiempo de intervalos especificado en las columnas de detalles de configuración de programa “Scan fijo” (fixed scan) y “Unidades de entrada” (In units) de la ventana de diálogo del PLC Parameter. (Ver sección 4.2.5.)

(c) Tiempo de scan de programas en ejecución (excepto el programa de tipo de ejecución de scan fijo) puede ser comprobado en el monitoreo listado de programa. (Ver sección 7.11.1.)

(2) Flujo de cada programa de alto nivel del modelo QCPU de altas prestaciones

El flujo de cada programa después del conmutado a ON o STOP del PLC a RUN del módulo CPU se muestra a continuación.


Commutado de ON/STOP a RUN
*
Programa tipo de ejecución inicial
Ejecutar solamente una vez se commuta a ON o se pasa de STOP a RUN
*
Programa de tipo de ejecución de baja velocidad
Activar solamente cuando el “scan constante” o “el tiempo de ejecución de programa a baja velocidad” ha sido configurado
Final del proceso
*
Programa del tipo de ejecución de scan prefijado
Programa activado a un ciclo especificado de intervalos de tiempo
*
Program de tipo ejecución scan
*
Programa del tipo Stand-by
Activar solo cuando se da la petición de ejecución

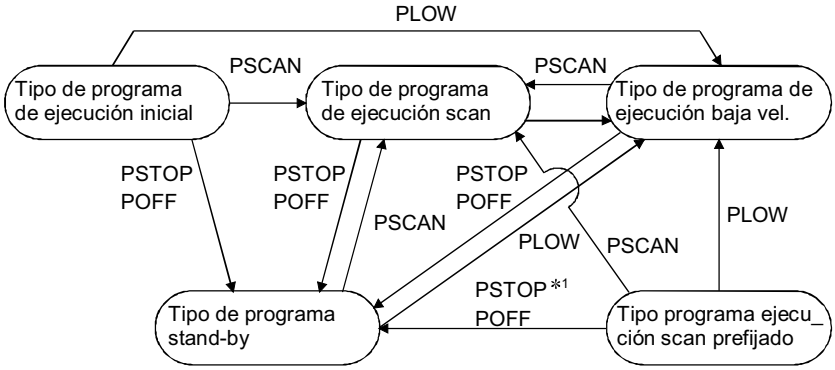
Comentarios

No todos los tipos de ejecución necesitan ser configurados para el modelo QCPU de altas prestaciones.

Usar los valores marcados con "*" necesarios, tales como la ejecución inicial, ejecución a baja velocidad, stand-by y programas de tipo de ejecución de scan fijo.

(3) Cambiando el tipo de ejecución

(a) La configuración del tipo de ejecución realizada en la pestaña "Program" pestaña en la ventana de diálogo “(PLC) Parameter" puede ser cambiada en cualquier momento mientras una secuencia de programa se ejecuta. Para cambiar la ejecución del tipo de programa, usar las instrucciones PSCAN, PLOW, PSTOP o POFF.



(b) La siguiente tabla muestra los tiempos de cambio de tipo de ejecución de un programa usando instrucciones PSCAN, PLOW, PSTOP y POFF.

Instrucción ejecutada / Tipo de ejecución antes de cambio	PSCAN	PSTOP	POFF	PLOW
Tipo de ejecución scan	No cambia – Permanece el tipo de ejecución scan	Pasa a tipo stand-by.	La salida pasa a OFF en el siguiente scan. Pasa a tipo stand-by desde el siguiente scan después de todo.	Pasa a tipo de baja velocidad.
Tipo de ejecución inicial	Pasa a tipo de ejecución scan.			
Tipo Stand-by		No cambia permanece en tipo stand-by.	No procesa.	
Tipo de ejecución de baja velocidad	Ejecución del tipo ejecución de baja velocidad es deshabilitado. Pasa a ejecuciones scan desde el siguiente scan. (Ejecución desde el paso 0).	Tipo de ejecución de ejecución de baja velocidad es deshabilitado. Pasa a tipo stand-by desde el siguiente scan.	Tipo de ejecución de ejecución de baja velocidad es deshabilitado y la salida pasa a OFF en el siguiente scan. Pasa a tipo stand-by desde el siguiente scan después de todo.	No cambia – permanece en ejecución low speed.
Tipo de ejecución de scan fijo	Pasa a tipo de ejecución stand-by	Pasa a tipo de ejecución stand-by	La salida cambia a OFF en el siguiente scan. Pasa a tipo stand-by type desde el siguiente scan después de todo,	Pasa a tipo de baja velocidad.

Comentarios

*1: Si la ejecución del tipo de programa de scan fijo se cambia por otro tipo de ejecución, no se podrá retornar al tipo de ejecución de scan fijo.

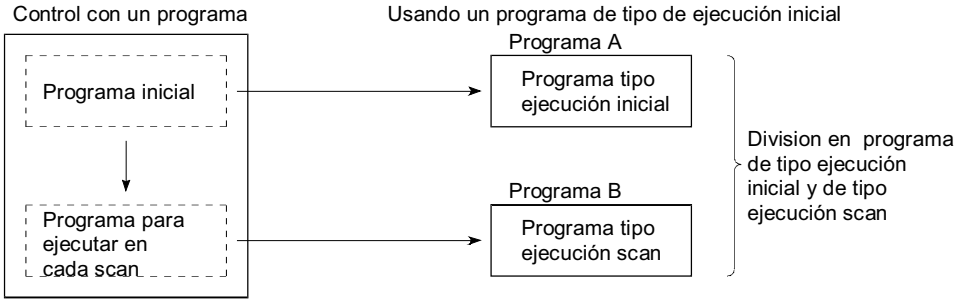
4.2.1 Definición de programa de tipo de ejecución inicial

(1) Definición de programa de tipo de ejecución inicial

(a) Un programa de tipo de ejecución inicial se ejecuta solamente si se conmuta a ON, o cuando se pasa de STOP a RUN.

(b) Esta ejecución de tipo de programa se designa como “inicial” en el programa en “PLC parameters”.

(c) De la misma forma como en el proceso inicial para el módulo de función inteligente, el programa de ejecución inicial es ejecutado solamente una vez y no es requerido en los posteriores ciclos de scan.
Una instrucción que contenga un dispositivo completo no puede ser usada para un tipo de programa de ejecución inicial porque el dispositivo completo necesita varios scan para completar la ejecución.

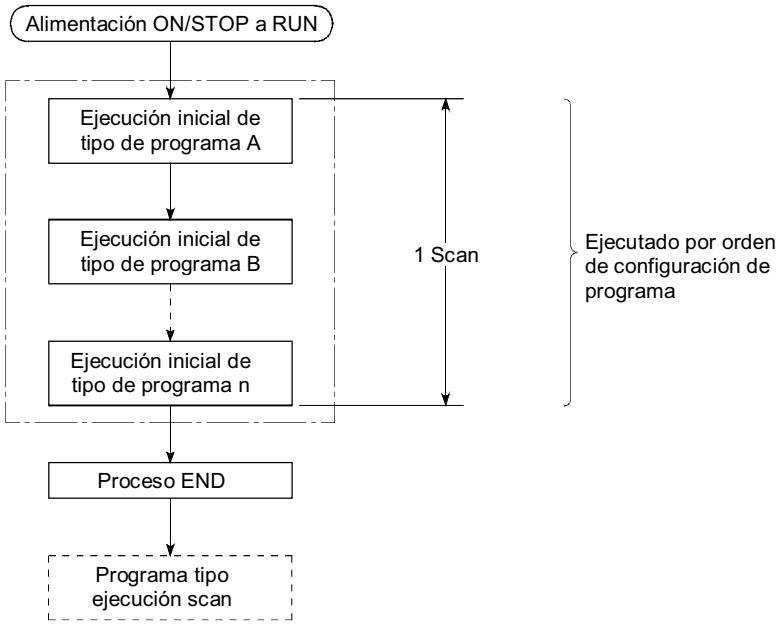

Control con un programa
Usando un programa de tipo de ejecución inicial
Programa A
Programa inicial
Programa tipo ejecución inicial
Division en programa de tipo ejecución inicial y de tipo ejecución scan
Programa B
Programa para ejecutar en cada scan
Programa tipo ejecución scan

(2) Usando múltiples programas de tipo ejecución inicial

Cuando múltiples programas de ejecución inicial son usados, serán ejecutados uno por uno en sentido ascendente del programa en el listado de “PLC parameters”.

(3) Proceso END

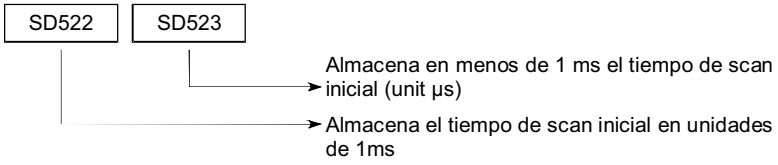
El proceso END actúa cuando todos los programas de tipo ejecución inicial son completados, y el “programa de tipo ejecución scan” es entonces ejecutado desde el siguiente scan.


Alimentación ON/STOP a RUN
Ejecución inicial de tipo de programa A
Ejecución inicial de tipo de programa B
Ejecución inicial de tipo de programa n
1 Scan
Ejecutado por orden de configuración de programa
Proceso END
Programa tipo ejecución scan

(4) Tiempo de scan inicial

(a) Este es el periodo de tiempo de ejecución para programas de tipo de ejecución inicial.
Si son ejecutados varios programas de ejecución inicial, este es el tiempo en el cual todos esos programas son ejecutados.
Cuando una interrupción de programa o programa de tipo de ejecución de scan fijo es ejecutado mientras un programa de ejecución inicial está ejecutando, el tiempo de ejecución del programa de interrupción / ejecución programa de scan fijo será añadido al programa de tipo de ejecución inicial.

(b) El modelo QCPU de altas prestaciones mide el tiempo de scan inicial y almacena el resultado en registros especiales (SD522, SD523). ∗1
El tiempo de scan inicial puede ser chequeado por monitorización de los registros especiales SD522 y SD523.


SD522
SD523
Almacena en menos de 1 ms el tiempo de scan inicial (unit µs)
Almacena el tiempo de scan inicial en unidades de 1ms

Si el valor de SD522 es 3 y el valor de SD523 es 400, el tiempo de scan inicial es de 3.4 ms.

Comentarios

∗1: La precisión del tiempo de scan inicial almacenado en los registros especiales es de ± 0.1 ms.
El contaje del tiempo de scan inicial continuará incluso si se ejecuta un reset del watchdog (WDT) en la secuencia de programa.

(5) Tiempo de monitoreo de ejecución inicial

(a) El periodo de ejecución de program de ejecución inicial puede ser monitorizado por este temporizador. El valor por defecto no está configurado.
Cuando se monitoriza el tiempo de ejecución del programa de ejecución inical, asignar el tiempo de monitoreo entre el rango de 10 a 2000 ms en la pestaña “PLC RAS" en la ventana de diálogo “(PLC) Parameter" .
(Configurando la unidad: 10 ms)

(b) El programa de ejecución en baja velocidad se ejecuta después de que se completa la ejecución del programa de ejecución inicial.
Para usar el programa de ejecución de baja velocidad, especificar el tiempo que es mayor que la suma del tiempo de scan inicial y el tiempo de ejecución del programa de ejecución de baja velocidad.

(c) Cuando el tiempo de scan inicial excede del tiempo de monitoreo de ejecución inicial , ocurre un "WDT ERROR (código de error: 5000)" y el funcionamiento del modelo QCPU de altas prestaciones se para.

Comentarios

Cuando el tiempo de monitoreo de ejecución inicial es designado, habrá un error de 10 ms en el valor de contaje.
Por lo tanto, una configuración de tiempo de monitoreo (t) de 10 ms causará "WDT ERROR" cuando el tiempo de scan inicial se encuentra en el rango comprendido 10 ms < t < 20 ms.

4.2.2 Programa de ejecución por scan

(1) Definición de programa de ejecución por scan

(a) Los programas de ejecución por scan son ejecutados una vez por scan, empezando desde el scan que sigue la ejecución del programa de ejecución inicial.

(b) Configurar el tipo de ejecución "scan" en la pestaña "Program" en la ventana de diálogo “(PLC) Parameter".

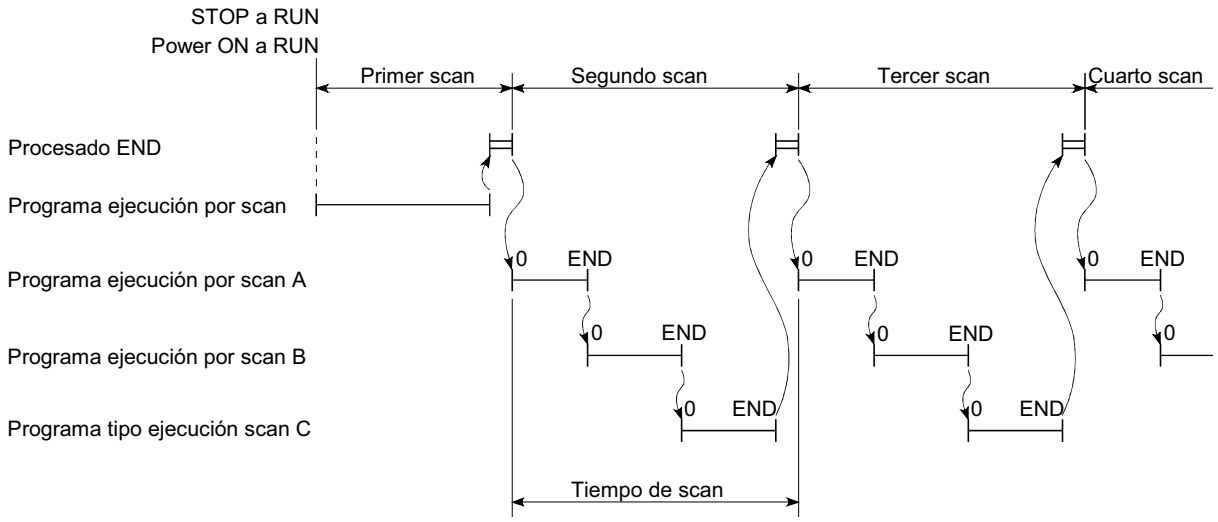
(2) Ejecutando múltiples programas de ejecución por scan múltiple

Cuando múltiples programas de ejecución por scan son usados, se ejecutan uno por uno en orden ascendente configurados en la pestaña “Program tab” en la ventana de diálogo “(PLC) Parameter".

(3) Procesado de END

El primer programa de ejecución por scan es ejecutado otra vez, cuando todos los programas de tipo ejecución scan son ejecutados y el procesado END es completado.

El procesado END (refresco de red) puede ser representado por cada programa cuando varios programas de ejecución por scan son ejecutados. Para hacer esto, incluir una instrucción COM al final de cada programa de ejecución por scan.



(4) Scan Constante

Un scan constante es una función que repite la ejecución de programa de ruta principal en los intervalos dados.

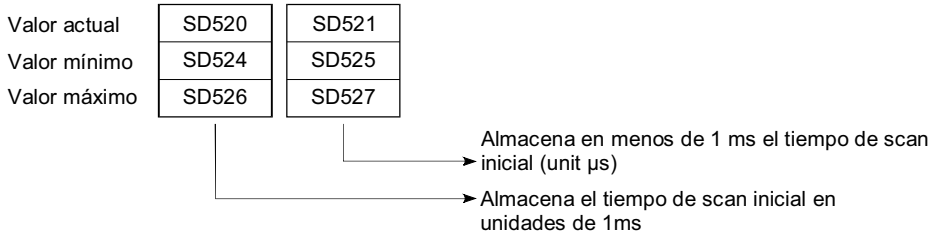
Cuando se configura un scan constante, el programa de ejecución por scan es ejecutado en los intervalos fijos de tiempo de scan constante de reset.

Referirse a la sección 7.2 para detalles acerca del scan constante.

(5) Tiempo de scan

(a) El “tiempo de scan” es un total de siguiendo la ejecución de tiempo del programa de ejecución por scan y procesado END.
SI múltiples programas de tipo ejecución por scan son usados, el “tiempo scan” es el tiempo total requerido para ejecutar todos los programas.
Cuando una interrupción de un programa de ejecución por scan fijo es ejecutada, el valor añadido al tiempo de ejecución del programa de ejecución por scan fijo o programado pasará a ser el tiempo de scan.

(b) Se promedia el valor actual de tiempo de scan, valores mínimo y máximo en el modelo de alto nivel QCPU y los resultados son almacenados en registros especiales (SD520, SD521, y SD524 a SD527). ∗1
Por lo tanto, el tiempo de scan inicial puede ser chequeado por monitorizado de los registros especiales SD520, SD521 y de SD524 a SD527.



Si el valor de SD520 es 3 y valor de SD521 es 400, el tiempo de scan inicial es de 3.4 ms.

Comentarios

∗1: La exactitud del tiempo de scan almacenado en los registros especiales es de ± 0.1 ms. El contaje del tiempo de scan continuará incluso si la se resetea el temporizador watch dog (WDT) es ejecutada en la secuencia de programa.

(6) WDT (Watch dog timer/ perro guardián)

Este es el temporizador que monitoriza el tiempo de scan y su configuración por defecto es de 200 ms.
Esta configuración WDT puede ser designada entre el rango de 10 a 2000 ms con la pestaña "PLC RAS" pestaña en la ventana de diálogo “(PLC) Parameter". (Unidad de configuración: 10 ms)
Cuando se usan programas de tipo de ejecución de baja velocidad, configurar el WDT superior que el tiempo de scan más el tiempo de ejecución del programa de tipo de ejecución de baja velocidad.
Si el tiempo de scan (tiempo de ejecución para programas de ejecución scan + programas de ejecución de baja velocidad) excede el valor de configuración WDT, ocurrirá un "WDT ERROR (código de error: 5000)" y se pararán las operaciones del modelo QCPU de altas prestaciones.

Comentarios

La medida de error del WDT es de 10 ms.
Por lo tanto, una configuración de WDT (t) de 10 ms resultará en "WDT ERROR" si el tiempo de scan se encuentra en el rango: 10 ms < t < 20 ms.

Observaciones

Usar el listado monitorizado de programa de GX Developer para comprobar el tiempo en que el programa está siendo ejecutado. Ver sección 7.11.1 para más detalles acerca del listado monitorizado de GX Developer.

4.2.3 Programa de ejecución a baja velocidad

(1) Definición de programa de ejecución de baja velocidad

(a) Los programas de ejecución de baja velocidad son ejecutados solamente durante “el tiempo sobrante de scan constante” o durante el periodo designado por “el tiempo de ejecución de programa de baja velocidad”.

1) Para un tiempo de scan constante con mayor exactitud, designar una configuración de tiempo de scan constante en la pestaña "PLC RAS" en “(PLC) Parameter".
(Rango de ajuste: 0.5 a 2000 ms, unidad de configuración: 0.5 ms)

2) Para asegurar el tiempo de ejecución para programas de tipo de ejecución de baja velocidad en cada ciclo de scan, designar un tiempo de ejecución de programa de baja velocidad en la pestaña "PLC RAS" del (PLC) "Parameter".
(Rango de configuración: 1 a 2000 ms, unidad de configuración: 1 ms)

3) Para ejecutar programas de ejecución de baja velocidad, configurar el tiempo de scan constante o el tiempo de ejecución de programa de baja velocidad.

(b) Configurar el tipo del programa de baja velocidad a “low speed” en el programa del ”PLC parameters”

(c) El programa de ejecución de baja velocidad es usado por programas que no requieren ejecución en cada scan, por ejemplo programas para salida de impresora.

(2) Ejecutando múltiples programas de tipo de ejecución de baja velocidad

Cuando son usados múltiples programas de tipo de ejecución de baja velocidad, son ejecutados uno por uno en orden numérico ascendente de los programas en el “PLC parameters”.

(3) Tiempo de Ejecución del programa de ejecución de baja velocidad para ser ejecutado por scan

(a) Si toda operación de programa de ejecución de baja velocidad es completada entre un scan y hay un tiempo sobrante, la ejecución del procesado, posterior, depende del estado ON/OFF del registro especial SM330 y la condición de ejecución para programas de tipo de ejecución de baja velocidad.

1) Método asíncrono (SM330 = OFF)
Método en el cual el funcionamiento de programa sobrante de ejecución de baja velocidad prosigue durante el tiempo sobrante.

2) Método síncrono (SM330 = ON)
Método en el cual incluso si hay un tiempo sobrante, el funcionamiento del programa de ejecución de baja velocidad prosigue y el funcionamiento empieza otra vez desde el siguiente scan.

Método de funcionamiento para programas de ejecución de baja velocidad	Estado de configuración SM330	Condición de ejecución de programas de ejecución de baja velocidad	
		Cuando se configura un tiempo de scan constante	Cuando se configura un tiempo de ejecución de programa de baja velocidad
Método asíncrono	OFF	El programa de ejecución de baja velocidad es re-ejecutado *1.	El programa de ejecución de baja velocidad es re-ejecutado *1.
Método síncrono	ON	Se genera un tiempo de espera de scan constante *3.	Se inicia el funcionamiento del programa de ejecución scan *4.

*1 Si ha sido designado un tiempo de scan constante, el programa de ejecución de baja velocidad será ejecutado repetidamente durante el tiempo sobrante de scan constante.
Por lo tanto, el tiempo de ejecución de programa de ejecución de baja velocidad varia de scan a scan.
Como en los programas de ejecución de baja velocidad, no se ejecutará en absoluto si el valor de tiempo sobrante de scan es de 0.5ms o inferior, debería ser designada una configuración de tiempo de scan que proporcionará un tiempo de sobrante de más de 0.5 ms.

*2 Si ha sido designado un tiempo de ejecución de programa de baja velocidad, el programa de ejecución de baja velocidad será ejecutado repetidamente de acuerdo con la configuración de tiempo.
Por lo tanto, el tiempo de scan variará de scan a scan.

*3 Si un tiempo de scan constante ha sido designado, el tiempo sobrante después del completado el procesado de baja velocidad END es el tiempo de espera y la ejecución del programa de tipo de ejecución scan empieza cuando el tiempo de scan constante ha transcurrido.
Esto significa que el tiempo de scan es constante en cada scan.
Sin embargo, si el tiempo de acarreo después del scan constante es menor que 0.5 ms, los programas de ejecución de baja velocidad no podrán ser ejecutados. Si usamos un programa de ejecución de baja

velocidad, configurar el tiempo de scan constante de forma que el tiempo de acarreo sea de 0.5 ms o mayor.

*4 Si ha sido designado un "tiempo de ejecución de programa de baja velocidad", se inicia el funcionamiento de programa de ejecución scan ignorando el tiempo sobrante después de todo el procesado END de baja velocidad.
Esto significa que el tiempo de scan difiere en cada scan.

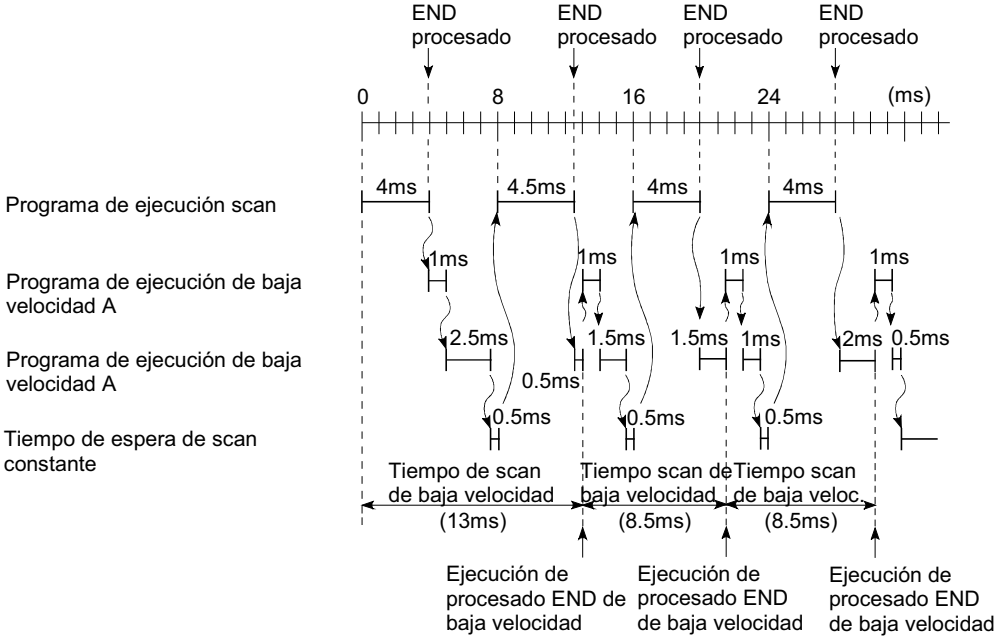
(b) Si un programa de ejecución de baja velocidad no puede ser procesado dentro de un tiempo sobrante de scan constante o dentro del tiempo de ejecución de programa de ejecución de baja velocidad, la ejecución del programa parará temporalmente y el resto del programa será ejecutado en el siguiente ciclo de scan.

1 : Método Asíncrono

(1) Configuración del tiempo de scan constante

El programa de ejecución de baja velocidad funciona según las siguientes condiciones como se muestran a continuación.

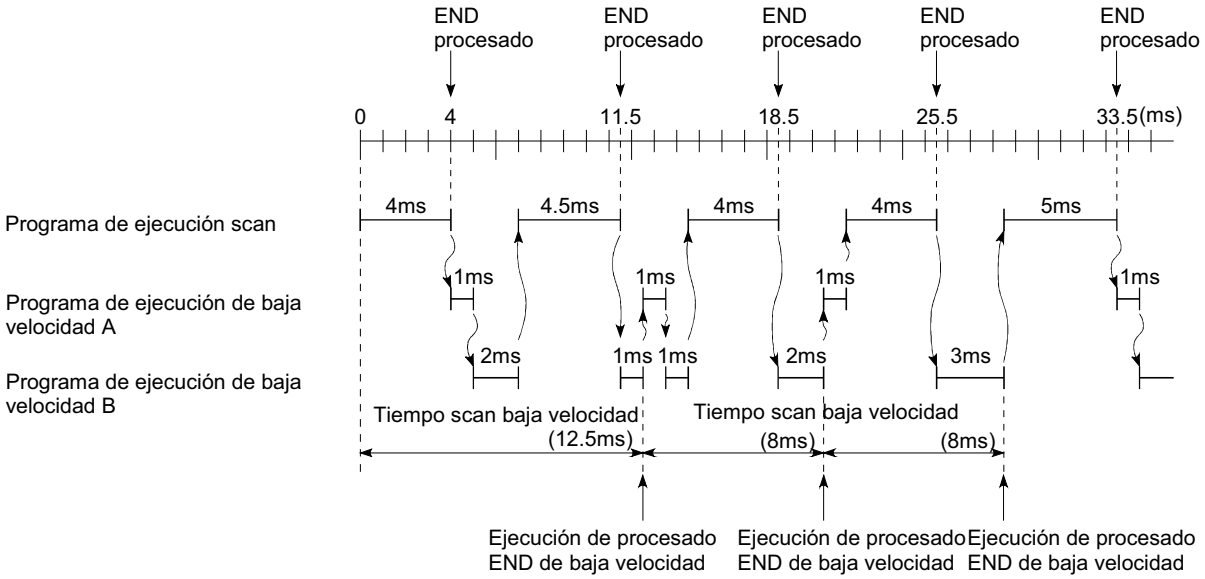
- Tiempo de scan constante: 8ms
- Tiempo total de programación de ejecución scan: 4ms a 5ms
- Ejecución de tiempo de ejecución de baja velocidad programa A : 1ms
- Ejecución de tiempo de ejecución de baja velocidad programa B : 3ms
- Procesado END /procesado de baja velocidad END : 0ms (0 ms para simplificar la ilustración)



(2) Configuración del tiempo de ejecución de programa de baja velocidad

La ejecución programa de baja velocidad funciona según las siguientes condiciones mostradas a continuación.

- Tiempo de ejecución de programa de baja velocidad : 3ms
- Tiempo de programa de ejecución scan : 4ms a 5ms
- Tiempo de ejecución de programa de ejecución de baja velocidad A : 1ms
- Tiempo de ejecución de programa de ejecución de baja velocidad B : 3ms
- proceso END /proceso END de baja velocidad: 0ms (0 ms para simplificar la ilustración)

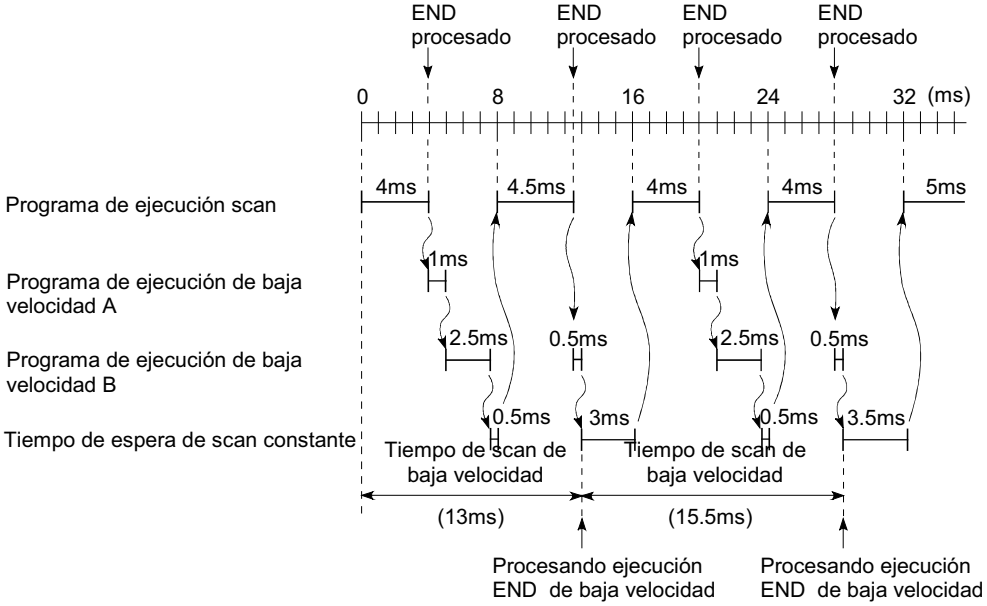


2 : Método síncrono

(1) Configuración del tiempo de scan constante

El programa de tipo de ejecución de baja velocidad funciona según las siguientes condiciones mostradas a continuación.

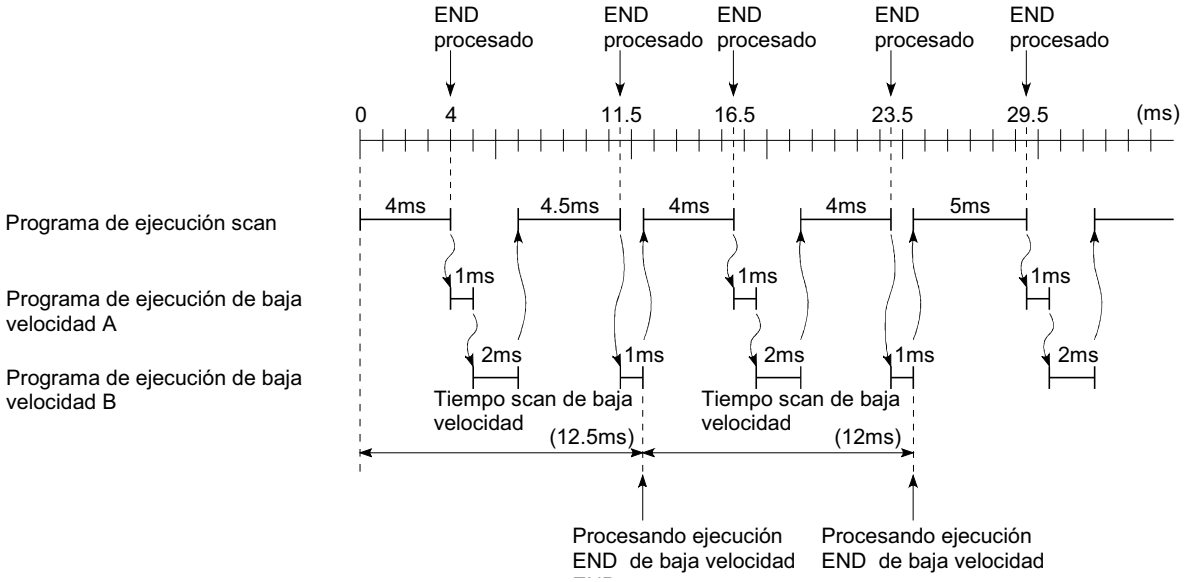
- Tiempo de scan constante : 8ms
- Tiempo total de programa de ejecución scan : 4ms a 5ms
- Tiempo de ejecución de programa de ejecución de baja velocidad A : 1ms
- Tiempo de ejecución de programa de ejecución de baja velocidad B : 3ms
- Procesado END/Procesado END de baja velocidad : 0ms (0 ms para simplificar la ilustración)



(2) Configuración del tiempo de ejecución del programa de baja velocidad

El programa de ejecución de baja velocidad funciona según bajo las siguientes condiciones mostradas a continuación.

- Tiempo de ejecución de programa de baja velocidad : 3ms
- Tiempo total de programa de ejecución scan : 4ms a 5ms
- Tiempo de ejecución de programa de baja velocidad A : 1ms
- Tiempo de ejecución de programa de baja velocidad B : 3ms
- Procesado END : 0ms (0 ms para simplificar la ilustración)



(4) Precauciones para la creación de programas de tipo de ejecución de baja velocidad

(a) Ver sección 10.6.1 para detalles sobre el procesado de registros índice cuando la conmutación desde un programa de ejecución scan a un programa de ejecución de baja velocidad.

(b) Ver sección 10.6.2 para detalles sobre el procesado de registros cuando una interrupción de programa o programa de ejecución de scan fijo se ejecuta durante la ejecución de un programa de ejecución de baja velocidad.

(c) El tiempo de ejecución de programa de baja velocidad debe ser configurado de forma que la suma del [tiempo de scan] + [tiempo de ejecución de programa de baja velocidad] es menor que el valor configurado en WDT.

(d) La instrucción COM puede no usarse en programas de ejecución de baja velocidad.

(e) Los programas de tipo de ejecución de baja velocidad pueden también ser ejecutados con scan que ejecuten los programas de ejecución inicial. Establecer un enlace entre SM402 y SM403 para el circuito que valide la operación de programas de ejecución de baja velocidad después de que el programa de ejecución de baja velocidad ha sido ejecutado.

(f) Cuando el "tiempo de scan constante" y "el tiempo de ejecución de programa de baja velocidad" han sido configurados, ocurrirá un "PRO. TIME OVER (código de error: 5010)" si el (tiempo sobrante de scan constante) < (tiempo de ejecución de programa de baja velocidad).

(5) Procesado END de baja velocidad

El procesado END de baja velocidad se efectúa cuando todos los programas de tipo de ejecución de baja velocidad son ejecutados.

EL siguiente procesado se efectúa por el procesado END de baja velocidad.

- Programa de baja velocidad de reles especiales / configurando registros especiales.
- Programas de ejecución de baja velocidad escrita durante RUN
- Medida de tiempo de scan de baja velocidad.
- Reseteo de temporizador watch dog de programa de tipo de ejecución de baja velocidad.

Cuando el procesado de baja velocidad END se completa, el programa de ejecución de baja velocidad se ejecuta desde el principio otra vez.

Comentarios

(1) Durante la ejecución de programas de ejecución de baja velocidad, el tiempo de scan constante puede desviarse mediante la cantidad de “El máximo tiempo de procesado de instrucciones + tiempo de procesado END de baja velocidad".

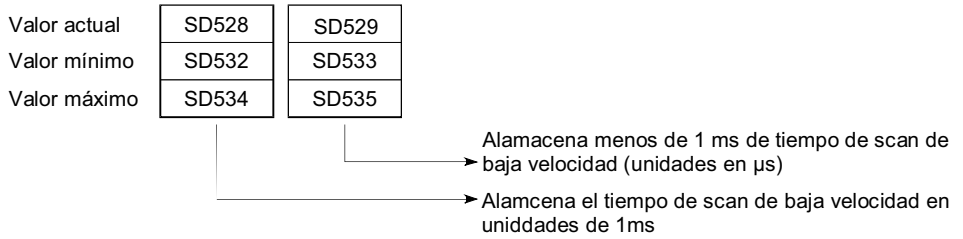
(6) Tiempo de scan de baja velocidad

(a) El tiempo de scan de baja velocidad es el tiempo total requerido para ejecución de programa de ejecución de baja velocidad y procesado END de baja velocidad.

Si múltiples programas de ejecución de baja velocidad son ejecutados, el tiempo de scan de baja velocidad es el tiempo total requerido para ejecutar todos los programas, además del tiempo de procesado END en baja

velocidad.
Cuando una interrupción de programa a programa de ejecución de scan prefijado se ejecuta, el valor añadido con la interrupción de programa / tiempo de ejecución de programa de ejecución por scan prefijado será el tiempo de scan de baja velocidad

(b) El tiempo de scan de baja velocidad es medido mediante el modelo QCPU de altas prestaciones y el resultado es almacenado en registros especiales (SD528 a SD535). *1
El tiempo de scan de baja velocidad puede ser por lo tanto chequeado mediante monitoreo de los registros especiales SD528 a SD535.



Si el valor de SD528 es 50 y el valor de SD529 es 400 el tiempo de scan de baja velocidad es 50.4 ms.

Comentarios

*1: La exactitud del tiempo de scan almacenado en los registros especiales es de ± 0.1 ms.
El tiempo de contaje de scan continuará incluso si la instrucción de reseteo de tiempo de watchdog (WDT) es ejecutada en la secuencia de programa.

(7) Tiempo de monitorización de ejecución de baja velocidad

El tiempo de ejecución de programas de ejecución de baja velocidad puede ser monitorizado por este temporizador. El valor por defecto no está configurado. Cuando se esté monitorizando el tiempo de ejecución del programa de tipo de ejecución de baja velocidad, designar el tiempo de monitoreo de ejecución de baja velocidad entre el rango de 10 y 2000 ms en la pestaña "PLC RAS" dentro de la ventana de diálogo “(PLC) Parameter". (Unidad de configuración: 10 ms)
Si el tiempo de ejecución de programa de ejecución de baja velocidad excede de este tiempo, ocurrirá un error "PRG TIME OVER".

Comentarios

La medida del tiempo de ejecución de baja velocidad ocurre en procesados END de baja velocidad.
Por lo tanto ocurrirá un error “PRG TIME OVER” si el tiempo de monitorización de ejecución de baja velocidad (t) es designado como 100 ms y la medida de tiempo de scan de baja velocidad en el procesado END de baja velocidad excede 100 ms.

4.2.4 Programa en Stand-by

(1) Definición de programa en stand-by

(a) Los programas en stand-by son ejecutados solamente cuando son solicitados.

(b) Los programas en Stand-by son usados para las siguientes aplicaciones.

1) Programas ubicados en la librería.

Subrutinas y programas de interrupción son convertidos en programas en stand-by los cuales son administrados separadamente desde el programa principal.

2) Cambiando el programa de setup.

Los programas de rutina principal registrados como programas en stand-by pueden ejecutar los programas requeridos para el control mediante su conversión a programas de ejecución por scan.

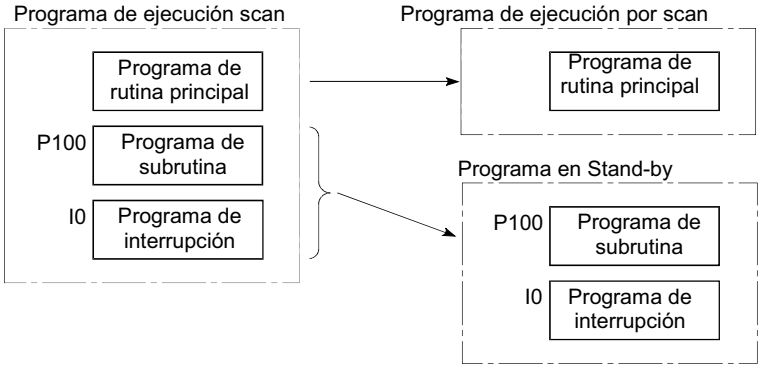
Serán reconvertidos a programas de tipo stand-by después de que completen la ejecución de programa.

(2) Aplicaciones de programa en Stand-by

(a) Ubicando programas en la librería

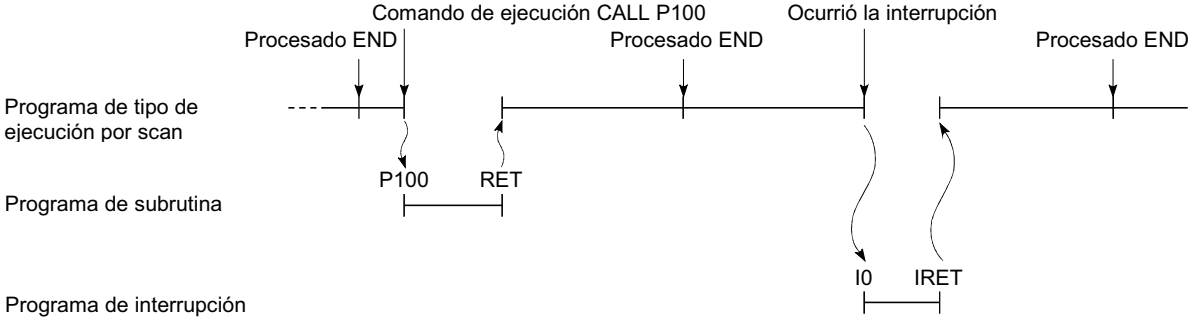
1) Esta aplicación es usada para administrar subrutinas y programas de interrupción separadamente desde el programa de interrupción principal.

Pueden crearse Subrutinas múltiples y programas de interrupción para programas en stand-by.



2) Cuando una ejecución de programa en stand-by es completada, el programa, el cual fue activado antes de que el programa en stand-by fuera ejecutado, será ejecutado.

Los programas de subrutina en stand-by y programas de interrupción son ejecutados tal y como como se muestra.



(b) Cambiando el programa de setup

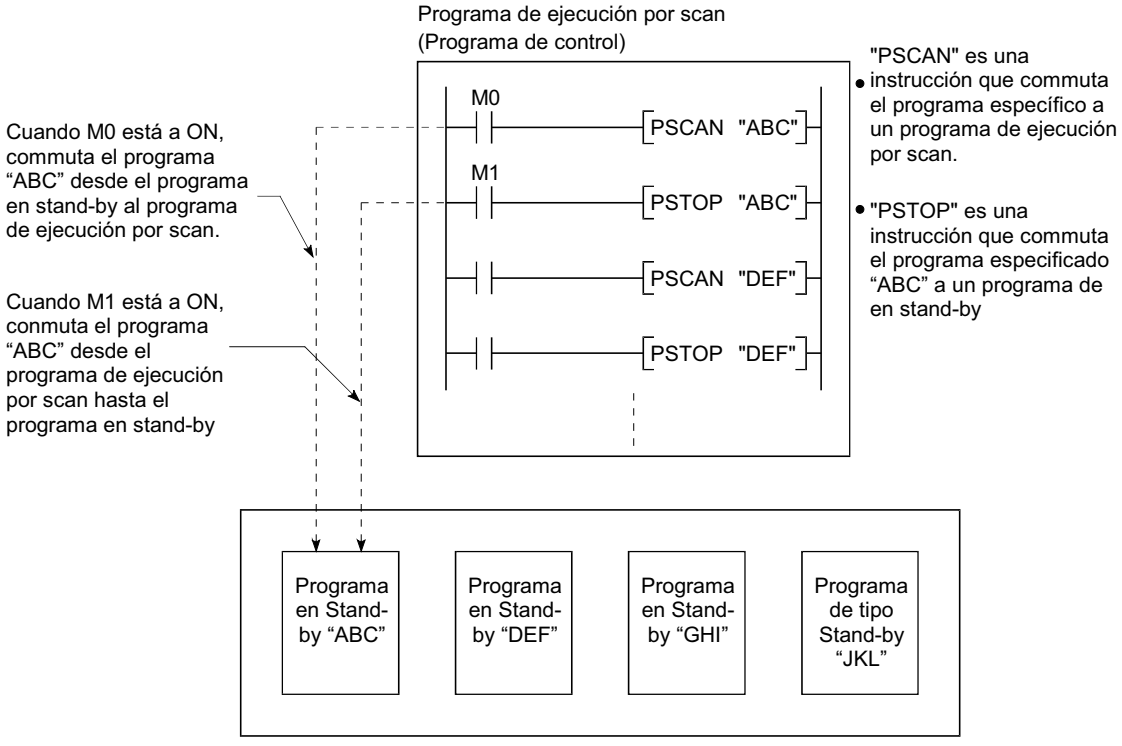
1) Crear un programa compatible con todos los programas y usarlo solamente para ejecutar programas necesarios.
Programas designados como programas en stand-by en que la ventana de diálogo “(PLC) Parameter" puede ser convertida a programas de ejecución scan y ejecutados en una secuencia de programa.
Cambiar el tipo de ejecución en el modelo QCPU de altas prestaciones mediante el uso de las instrucciones PSCAN, PLOW, PSTOP y POFF. (Ver sección 4.2 3).

(c) Los siguientes métodos no pueden ser usados para convertir un programa que va a ser ejecutado.

1) Seleccionando el programa para ser ejecutado desde un programa de gestión simple:

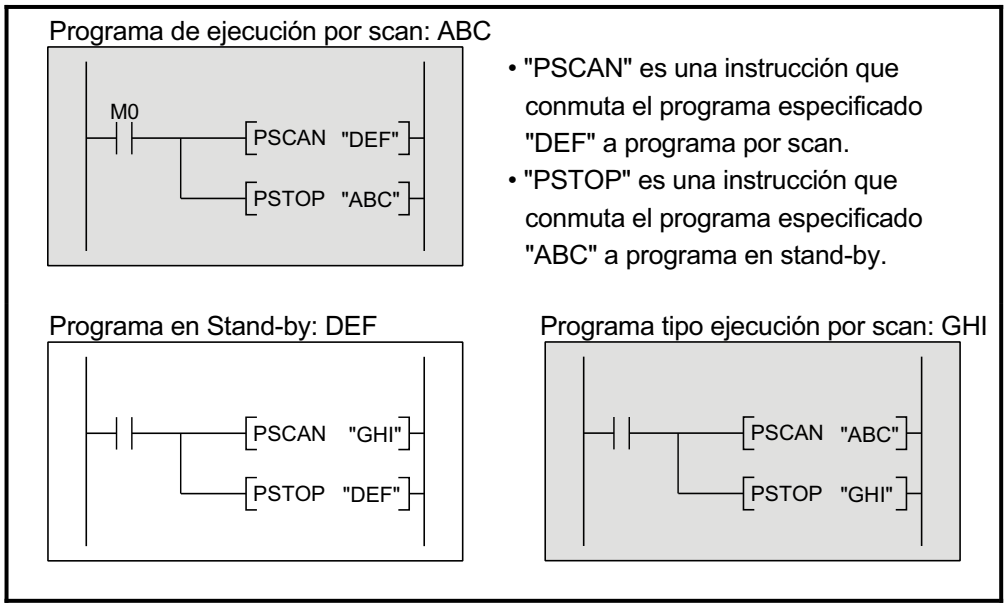
• Convertir un programa en stand-by que trate las condiciones designadas a un programa de ejecución por scan mediante el uso de la ejecutación constante de un programa de tipo de ejecución por scan como el programa principal. Luego ejecutar el programa convertido.
Los programas de ejecución por scan los que no son requeridos pueden ser convertidos a programas en stand-by.

• Los tipos de ejecución de programa en stand-by "ABC","DEF","GHI" y "JKL" son convertidos tal y como se muestra seguidamente.

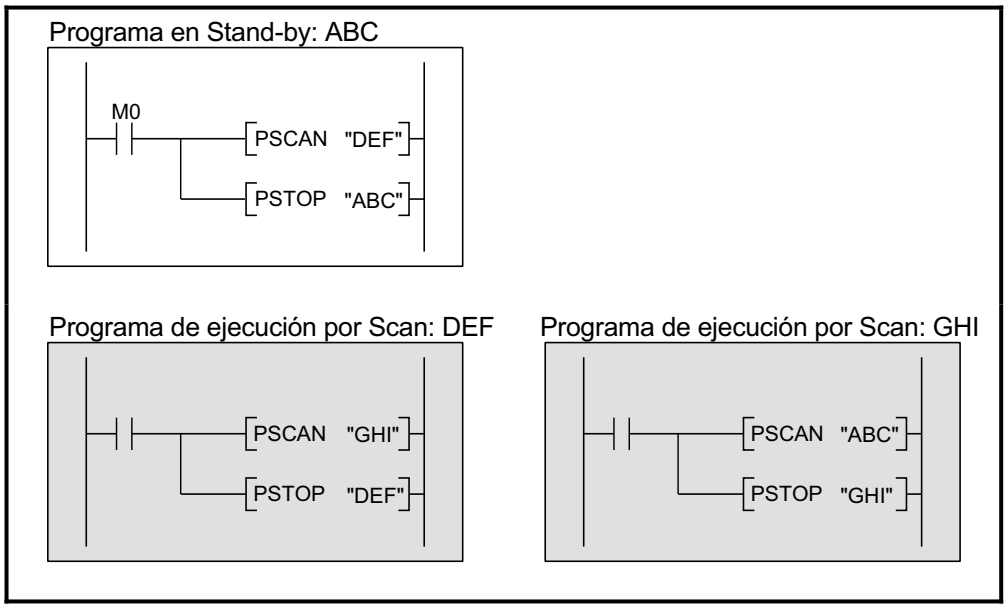

Programa de ejecución por scan
(Programa de control)
M0
PSCAN "ABC"
M1
PSTOP "ABC"
PSCAN "DEF"
PSTOP "DEF"
"PSCAN" es una instrucción que commuta el programa específico a un programa de ejecución por scan.
"PSTOP" es una instrucción que commuta el programa especificado “ABC” a un programa de en stand-by
Cuando M0 está a ON, commuta el programa “ABC” desde el programa en stand-by al programa de ejecución por scan.
Cuando M1 está a ON, conmuta el programa “ABC” desde el programa de ejecución por scan hasta el programa en stand-by
Programa en Stand-by “ABC”
Programa en Stand-by “DEF”
Programa en Stand-by “GHI”
Programa de tipo Stand-by “JKL”

2) Cuando cambiando el tipo de ejecución de los programas de tipo de ejecución por scan y los programas de tipo stand-by mediante los programas de ejecución por scan, teniendo la condición de conmutación de tipo de ejecución.
   • El programa de ejecución por scan que está siendo ejecutado cambia el siguiente programa a ser ejecutado desde el programa en stand-by a programa de ejecución por scan.
   • Si la condición se realiza cuando los programas “ABC” y “GUI” han sido configurados a ejecución por scan y el programa “DEF” en stand-by, el tipo de ejecución de los programas "ABC" y "DEF" son conmutados como se muestra seguidamente.

[Antes de la ejecución de las instrucciones PSCAN y PSTOP]

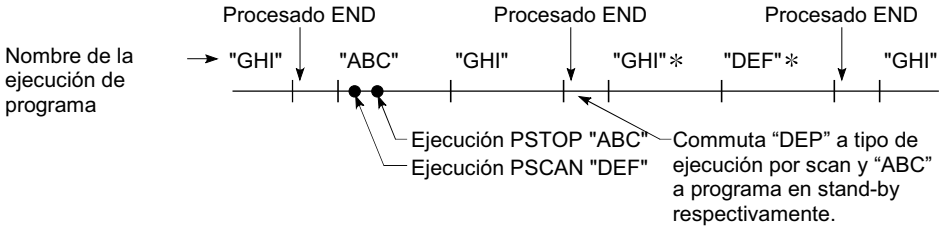

Programa de ejecución por scan: ABC
M0
PSCAN "DEF"
PSTOP "ABC"
• "PSCAN" es una instrucción que conmuta el programa especificado "DEF" a programa por scan.
• "PSTOP" es una instrucción que conmuta el programa especificado "ABC" a programa en stand-by.
Programa en Stand-by: DEF
PSCAN "GHI"
PSTOP "DEF"
Programa tipo ejecución por scan: GHI
PSCAN "ABC"
PSTOP "GHI"

[Después de la ejecución de las instrucciones PSCAN y PSTOP]



4 - 31

(d) El tipo de ejecución de programa es conmutado en el procesado END. El tipo de ejecución de programa no cambiará mientras el programa esté siendo ejecutado.
Si un tipo de ejecución diferente se especifica para el mismo programa en el mismo scan, el último tipo de ejecución especificada pasa a ser efectiva.



Comentarios

1) *: Los programas "GHI" y "DEF" son ejecutados en el orden como están configurados en la pestaña “Program” en la ventana de diálogo “(PLC) Parameter”.

(3) Precauciones para la creación de programas en stand-by

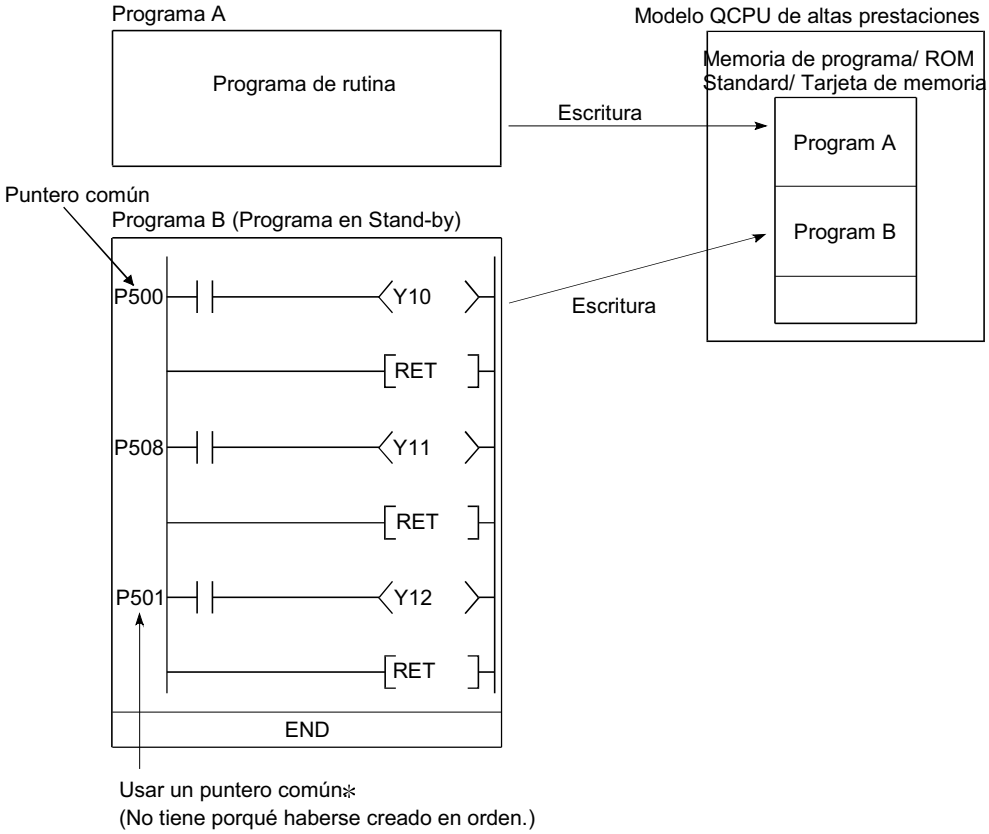
(a) Como el valor actual es actualizado y el contacto ON/OFF es conmutado cuando la instrucción OUT T⬚ es ejecutada, los temporizadores no pueden ser usados en programas en stand-by.

(b) Juntando programas de subrutina en programas simples

1) Crear los programas de subrutina en orden empezando desde el paso 0 del programa en stand-by. Se requiere una instrucción END al final del programa de subrutina.

2) Como no hay restricciones a cerca de la creación de programas de subrutina, el número de punteros no necesitan ser asignados en orden ascendente cuando se crean programas múltiples de subrutina.

3) Uso de punteros comunes. ∗
Puede llamarse programas de subrutina con punteros comunes desde todos los programas ejecutados mediante el modelo QCPU de altas prestaciones.
(Si son utilizados los punteros locales, los programas de subrutina en stand-by no serán ejecutados.)

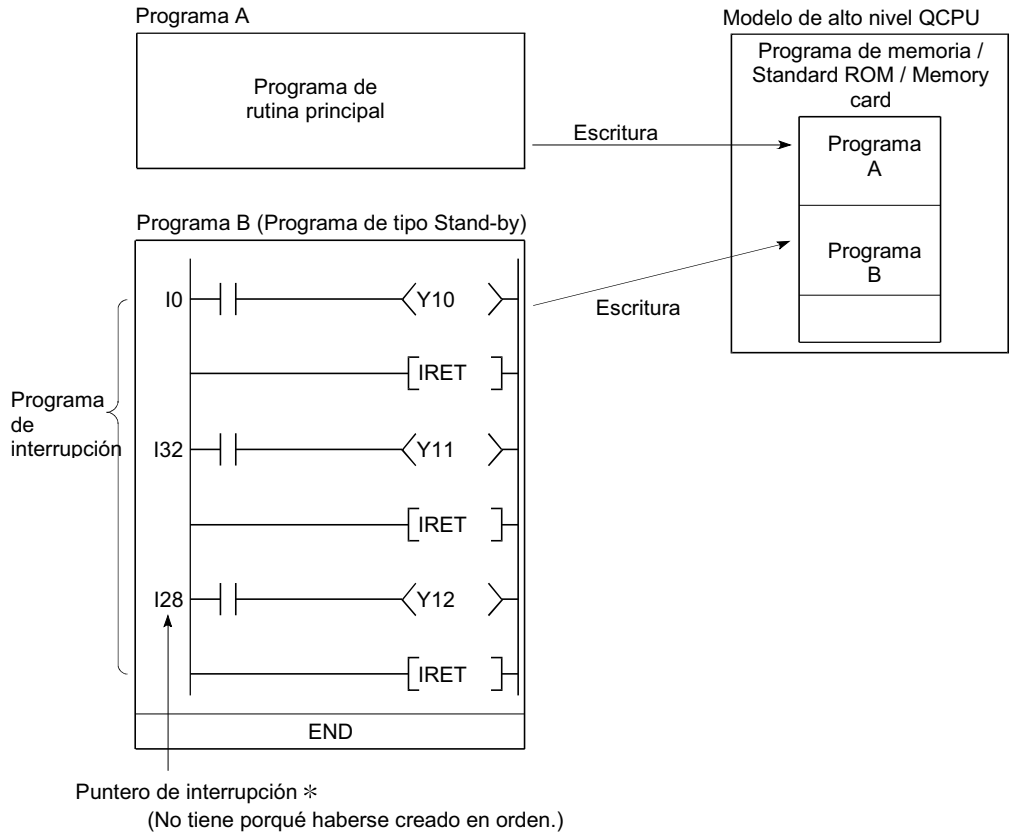


4) Ver sección 10.13.1 para ejecución de un programa de subrutina que contenga dispositivos locales.

(c) Juntando programas de interrupción en un programa simple.

1) Crear un programa de interrupción en orden empezando desde el paso 0 del programa en stand-by.
Una instrucción END no se requiere en el final de la interrupción de programa.

2) Como no hay restricciones en el orden de creación de programas de interrupción, el número de punteros no necesita ser asignado en orden ascendente cuando se están creando programas de interrupción múltiple.



4 - 34

4.2.5 Programa de tipo de ejecución scan fijo

(1) Definición de programa de tiempo de ejecución de scan fijo

(a) Este programa es ejecutado en los intervalos específicos.
Sin describir un puntero de interrupción y una instrucción de IRET, una ejecución de scan fijo puede ser representado en cada archivo.

(b) El tipo de ejecución es configurado a "Fixed Scan" en la pestaña de la ventana de diálogo del"(PLC) parameter".

(2) Ejecución de programa de ejecución scan fijo

(a) Los programas de ejecución de scan fijo son ejecutados en los intervalos específicos de tiempo de ciclo.
Cuando múltiples programas de ejecución scan fijo lo han alcanzado simultáneamente, son ejecutados en orden ascendente configurados en la pestaña de la ventana de diálogo “(PLC) Parameter"

(b) Configurar el intervalo periódico en el “programa” configurando la pestaña en la ventana de diálogo “(PLC) Parameter". La configuración del rango varia con la unidad configurada..
• Cuando la unidad es "ms": 0.5 a 999.5ms
• Cuando la unidad es "s": 1 a 60s

(c) Cuando los tiempos especificados de programas de ejecución scan fijo y las interrupciones de programa (I28 a I31) se han producido simultáneamente, la prioridad de la ejecución es dada a los programas de interrupción.

(d) Ejecución durante el refresco de red
Cuando las condiciones de ejecución de programas de tipo de ejecución scan fijo son establecidas durante el refresco de red, éste es suspendido y los programas de interrupción son ejecutados.
Por lo tanto, incluso si el “bloque de seguridad de datos cíclicos para cada estación” es realizado en el sistema de red MELSECNET/H, las operaciones superiores no se asegurarán cuando un dispositivo configurado para ser refrescado sea usado en el programa de interrupción ∗1

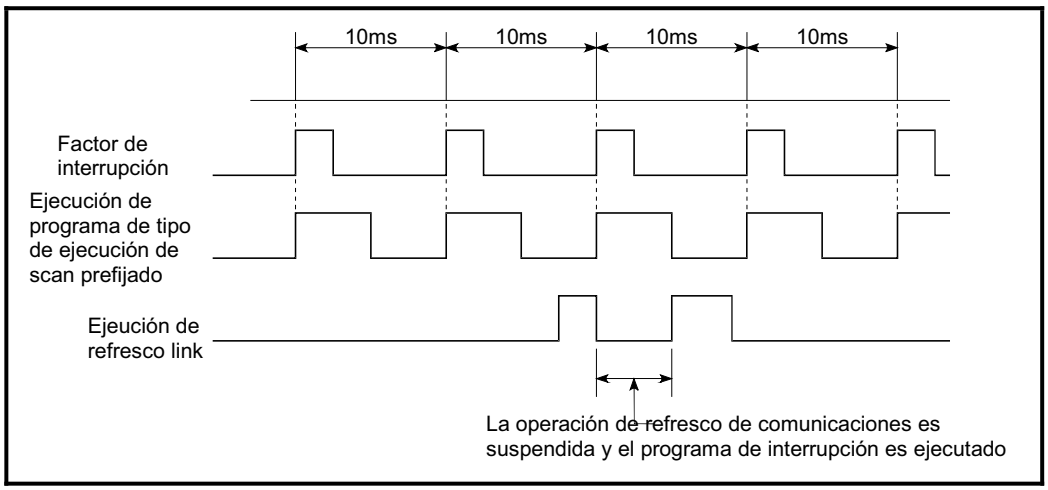

10ms
10ms
10ms
10ms
Factor de interrupción
Ejecución de programa de tipo de ejecución de scan prefijado
Ejeución de refresco link
La operación de refresco de comunicaciones es suspendida y el programa de interrupción es ejecutado

Fig. 4.5 Ejecución de programa de tipo de ejecución scan fijo durante el refresco de la red

Comentarios

*1: Referirse al manual adjunto dentro del bloque de seguridad de datos cíclicos para cada estación.
• Manual de referencia de sistema de red MELSECNET/H con correspondencia el modelo Q.

(e) Ejecución durante el procesado END:
Cuando las condiciones de ejecución de programa de ejecución scan fijo son establecidas durante el tiempo de espera de la instrucción END mientras el scan constante es ejecutado, los programas de ejecución scan fijo son ejecutados.

(f) Realizar el procesado del índice de registro cuando el programa es conmutado desde el programa de tipo de ejecución scan fijo a un programa de ejecución scan mediante la consulta de la sección 10.6.2.

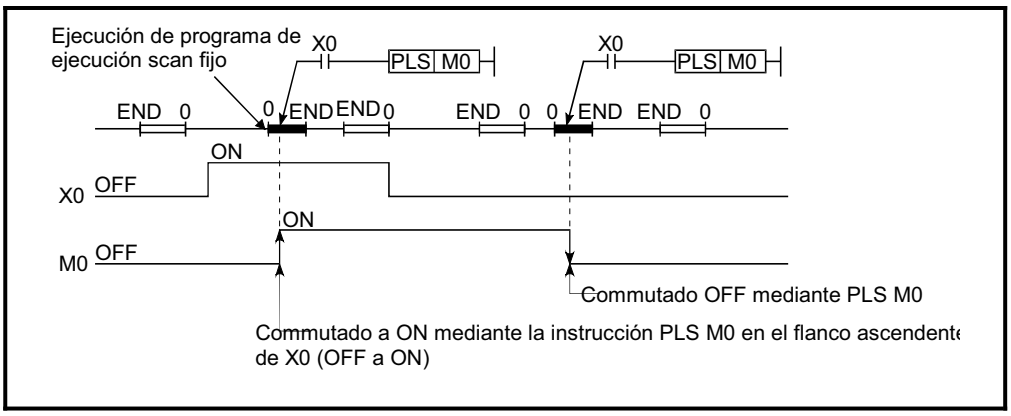
(3) Configuración de programa de scan fijo para ejecución de programas de ejecución de alta velocidad y tiempo de overhead

Cuando los programas de ejecución scan fijo son ejecutados, se realiza el procesado siguiente.
• Guardar y retornar el registro índice.
• Guardado y retorno del nombre de archivo del registro de archivo en uso.
Si la “Ejecución de alta velocidad” es seleccionada desde el programa de tipo de ejecución scan fijo en la pestaña “PLC system” en la ventana de diálogo “(PLC) Parameter", el procesado posterior no será representado.
Como resultado, el tiempo de overhead para los programas de ejecución de scan fijo puede ser reducido.

Tipo de CPU	Tiempo de Overhead	
	Ejecución de alta velocidad no seleccionada	Ejecución de alta velocidad seleccionada
Q02CPU	380 μ s	230 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	165 μ s	100 μ s

(4) Precauciones en la programación

(a) Si un dispositivo es conectado en un programa de ejecución scan mediante la instrucción PLS, es mantenido a ON hasta que el mismo tipo de programa de ejecución scan fijo es ejecutado otra vez.



4 - 36

(b) Durante la ejecución de un programa de tipo de ejecución de scan fijo, la interrupción es prohibida (DI). Por lo tanto, no ejecutar instrucciones EI/DI durante la programación de programas de tipo de ejecución de scan fijo.

(c) Durante la programación de programas de ejecución de scan fijo no se pueden utilizar temporizadores.
Como el temporizador actualiza el valor actual y conmuta a ON/OFF en el tiempo de ejecuciones de la instrucciones OUT T□, si el temporizador es usado durante la programación en el programa de ejecución de scan fijo, el valor actual será actualizado solamente cuando el programa de ejecución scan fijo sea ejecutado y la medida normal será deshabilitada.

(d) Los siguientes comandos no pueden ser usados en el programa de ejecución de scan fijo.
• COM
• ZCOM

(e) Cuando un programa de ejecución de scan fijo es ejecutado, debe permitirse las interrupciones mediante una instrucción EI en el inicio de la ejecución de programa de ejecución por scan.

(f) Cuando el programa de ejecución de scan fijo/ programa de interrupción es ejecutado y se mide el tiempo de scan o tiempo de ejecución, el valor de programa de interrupción/ programa de ejecución de scan fijo son añadidos a la medida de tiempo.
En definitiva, si el programa de interrupción / programa de ejecución de scan fijo es ejecutado, el valor almacenado en los siguientes registros especiales y el valor de monitoreo del GX Developer pasará a ser mayor que cuando el programa de interrupción/ programa de ejecución de scan fijo no es ejecutado.

1) Registros especiales
- SD520, SD521: Tiempo de scan actual
- SD522, SD523: Tiempo de scan inicial
- SD524, SD525: Tiempo de scan mínimo
- SD526, SD527: Tiempo de scan máximo
- SD528, SD529: Tiempo de scan actual para baja velocidad
- SD532, SD533: Tiempo de scan mínimo para baja velocidad
- SD534, SD535: Tiempo de scan máximo para baja velocidad
- SD540, SD541: Tiempo de procesado END
- SD542, SD543: Tiempo de espera de scan constante
- SD544, SD545: Tiempo de ejecución scan acumulativo para programas de baja velocidad
- SD546, SD547: Tiempo de ejecución de baja velocidad
- SD548, SD549: Tiempo de ejecución de programa scan
- SD551, SD552: Tiempo de intervalo de servicio

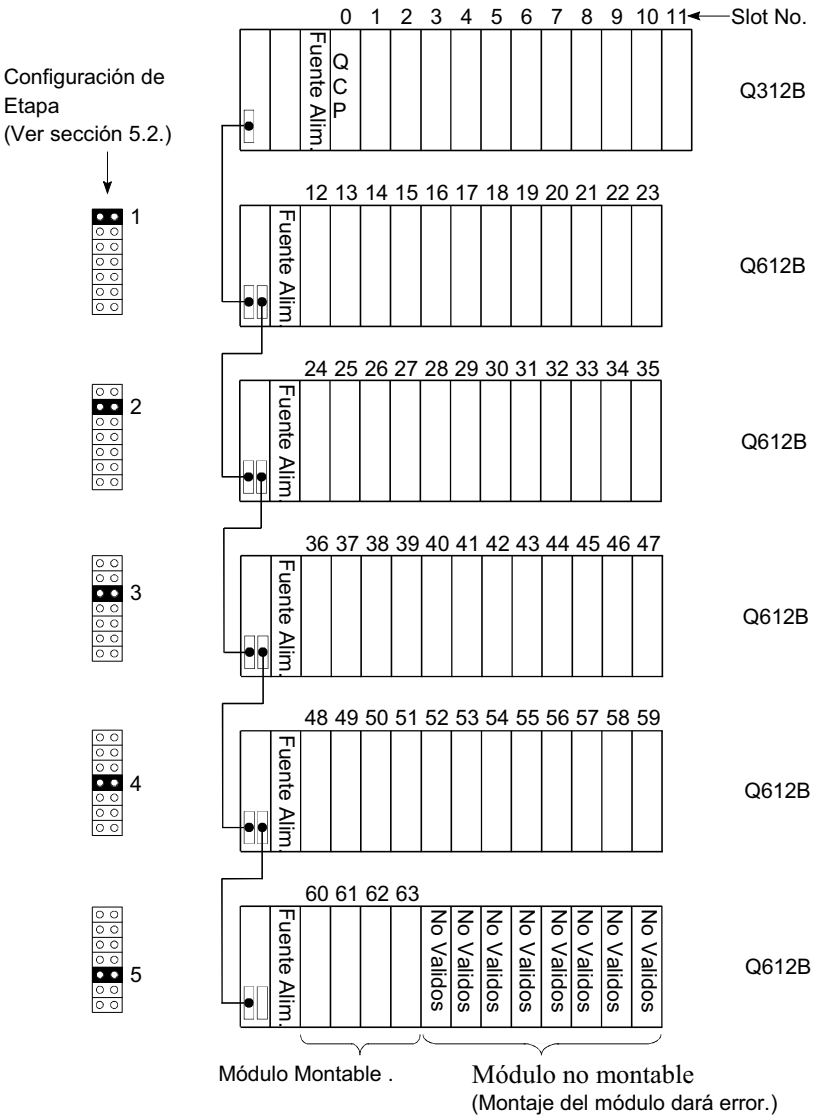
2) Valor de monitorización de GX Developer
- Tiempo de medida de ejecución
- Tiempo de medida de scan
- Scan constante

5 ASIGNACIÓN DE E/S

Esta sección describe la información necesaria para la asignación de E/S para el intercambio de información entre los modelos QCPU y los módulos de E/S o bien módulos con función inteligente.

5.1 Relación entre el número de puntos y los Slots de la Unidad Base de Extensión

La QCPU permite una configuración de sistema utilizando 8 unidades base: una unidad base principal y 7 bases de extensión.
No obstante, el número de slots admitidos (módulos) se limita a 64 incluyendo los slots vacíos.
Tiene lugar un error (SP. UNIT LAY ERR.) cuando un módulo (entrada, salida, o módulo con función inteligente) es instalado en el slot 65 o siguientes.
Asegúrese de instalar módulos dentro del rango de los 64 slots. (No debería ocurrir un error si los módulos son instalados en el rango de los 64 slots, aunque el número total de slots de la base principal y de las unidades de extensión resulta ser de 65 slots o más { Cuando son instaladas 6 unidades base 12-slots.)



5.2 Instalación de las Unidades Base de Extensión y configuración del número de puntos

Existen dos tipos de unidad base de extensión: Q5□B/Q6□B para el montaje de módulos de la serie Q y QA1S6□B para el montaje de módulos de la serie AnS .

(1) Orden de Conexión de las unidades base de extensión

Cuando utilizamos ambas Q5□B/Q6□B y QA1S6□B, conectar todos los módulos Q5□B/Q6□B cerca de la unidad base principal, a continuación conectar los módulos QA1S6□B .

(2) Configuración del orden de los puntos para las unidades base de extensión

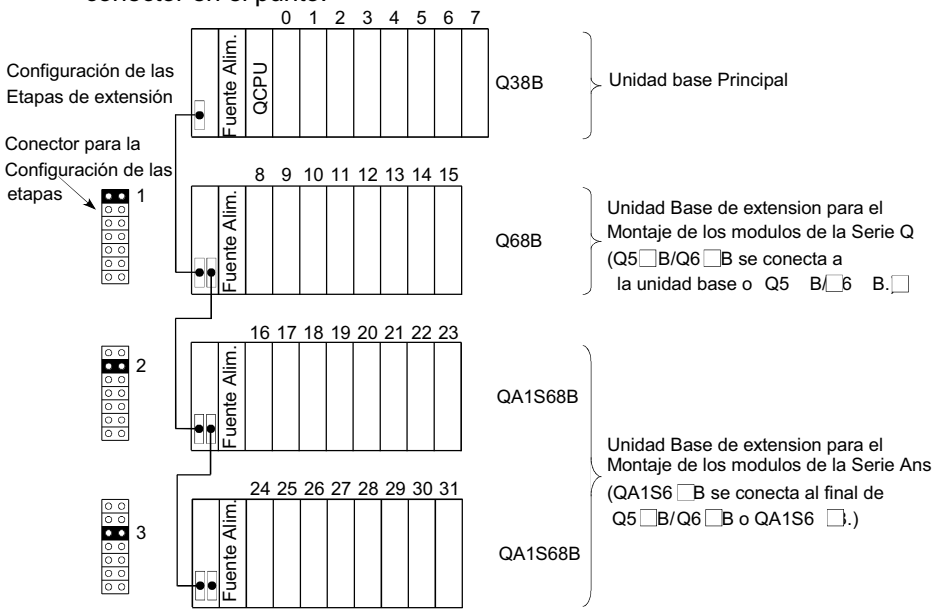
Las unidades base de Extensión necesitan la configuración de los puntos de extensión (1 a 7) utilizando el No punto en la configuración de conector.
Asigne el número de estación empezando desde el 1 al 7 de la unidad base de extensión en el orden de conexión empezando desde la conectada a la unidad base principal.

(3) Precauciones en la asignación de estaciones de extensión con la unidad base de extensión

(a) Asigne números consecutivos a las estaciones de extensión.
Si usted asigna un número de estación a la unidad base en modo "Auto" y asigna a alguna estación a no modulos, se asigna un "0" al punto saltado como slot. Por consiguiente, el número de slots vacios no incrementa. El punto saltado es también asignado con un “0” de punto E/S.

(b) Es imposible configurar y utilizar el mismo punto con 2 o más unidades base de extensión.

(c) Usted no puede utilizar el sistema si 2 o más conectores estan insertados en el conector del punto.
Por lo contrario, usted no puede utilizar el sistema si no hay insertado un conector en el punto.


0 1 2 3 4 5 6 7
Fuente Alim.
QCPU
Q38B
Unidad base Principal
Configuración de las Etapas de extensión
Conector para la Configuración de las etapas
1
8 9 10 11 12 13 14 15
Fuente Alim.
Q68B
Unidad Base de extension para el Montaje de los modulos de la Serie Q
(Q5□B/Q6□B se conecta a la unidad base o Q5 B/□6 B.□)
2
16 17 18 19 20 21 22 23
Fuente Alim.
QA1S68B
Unidad Base de extension para el Montaje de los modulos de la Serie Ans
(QA1S6□B se conecta al final de Q5□B/Q6□B o QA1S6 □.)
3
24 25 26 27 28 29 30 31
Fuente Alim.
QA1S68B

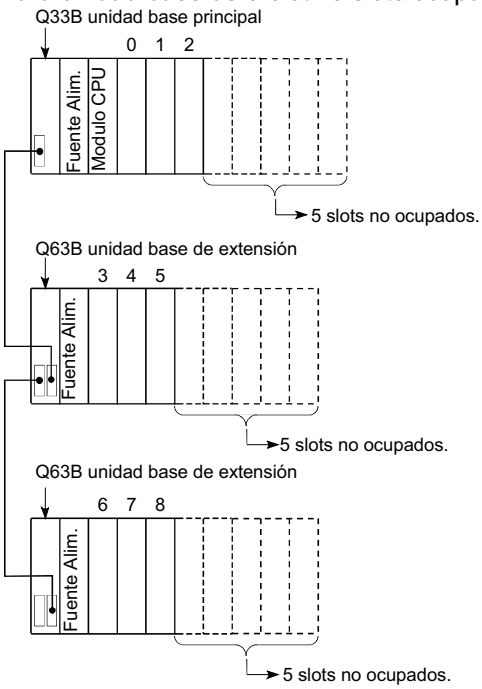
5.3 Asignación Unidad Base (Base Mode)

Existen los modos "Auto" y "Detail" para asignar el número de los módulos que pueden ser montados en la unidad base principal y las unidades base de extensión del modelo QCPU.

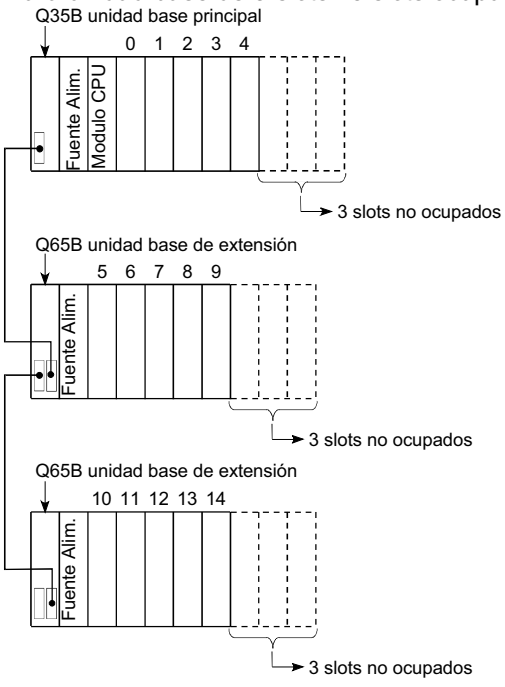
(1) Modo Auto

En modo Auto, los números de slots son asignados a la unidad base principal y las unidades base de extensión según el número de slots que pueden ser ocupados.
El número de E/S es asignado de acuerdo con los módulos que pueden ser montados a la unidad base actual.

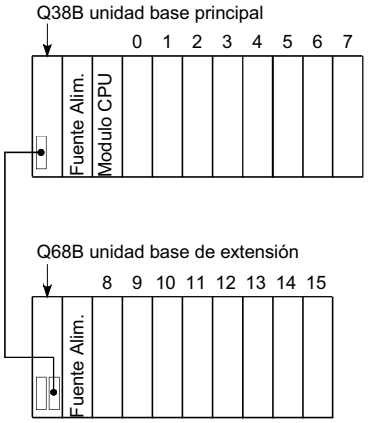
(a) Para unidad base de 3-slot: 3 slots ocupados



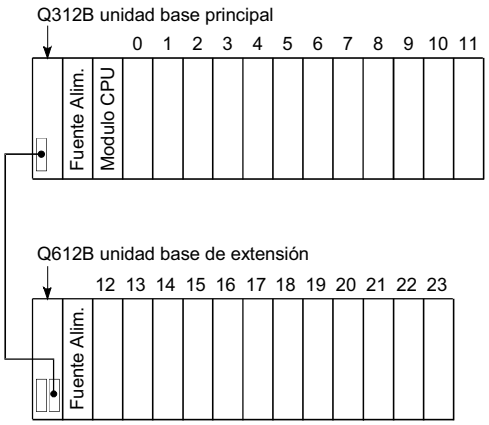
(b) Para unidad base de 5-slots: 5 slots ocupados



(c) Para unidad base de 8-slots: 8 slots ocupados



(d) Para unidad base de 12-slots: 12 slots ocupados

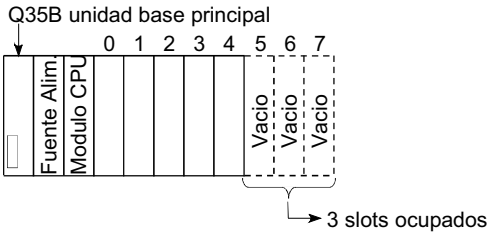

Q312B unidad base principal
0 1 2 3 4 5 6 7 8 9 10 11
Fuente Alim.
Modulo CPU
Q612B unidad base de extensión
12 13 14 15 16 17 18 19 20 21 22 23
Fuente Alim.

(2) Modo Detalle

(a) En modo Detalle, el número de modulos montables es asignado a las unidades base individuales (unidad base principal y unidad base de extensión) en la pestaña "E/S asignación" en el cuadro de diálogo “(PLC) Parameter" .
Use este modo para igualar el número de slots con los de la unidad base de la Serie AnS (8 fijaciones).

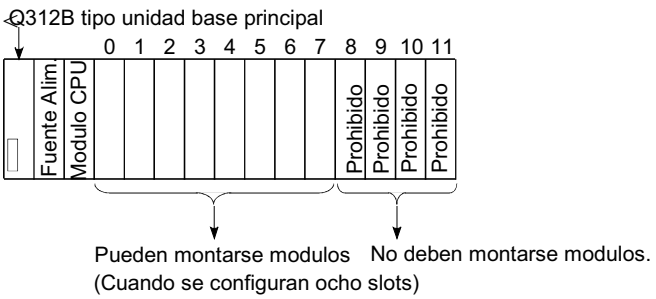
(b) Precauciones en la configuración del número de slots
El número de slots puede ser fijado sin tener en cuenta el número de módulos utilizados.
No obstante, el número de slots debe ser fijado para todas las unidades base en uso.
Si el número de slot no se define para todas las unidades base, la asignación E/S puede no funcionar correctamente.
Puede ocurrir lo siguiente si el número prefijado de los slots difiere de los instalados en la unidad base:

1) Cuando el número de slots designados es mayor que los instalados en la unidad base:
Entre los slots designados, aquellos después de los slots ocupados por la unidad base instalada serán slots vacios.
Por ejemplo, cuando se designan 8 slots para una unidad base de 5-slot, 3 slots serán slots vacios.

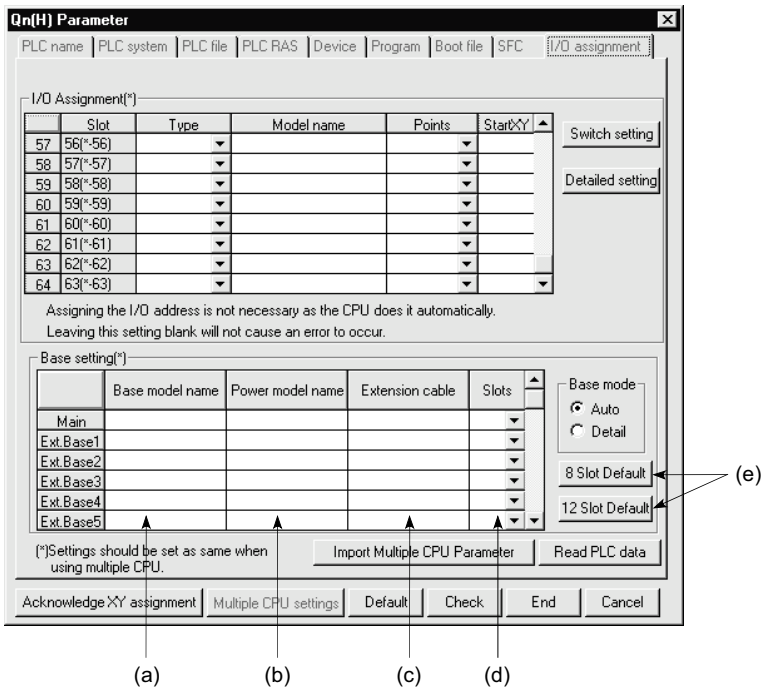


El número de puntos para los slots vacíos es el designado en la pestaña " asignación E/S " en el cuadro de diálogo “(PLC) Parameter" área de dialogo. (Valor por defecto 16 puntos.)

2) Cuando el número de slots designados es menor que los que estan en uso de la unidad base:
Los otros slots designados estan inhabilitados.
Por ejemplo, cuando 8 slots son designados para una unidad base de 12-slots, los 4 slots de la derecha de la unidad base quedan inhabilitados.(Si un modulo se monta en un slot prohibido, ocurre un error [SP. UNIT LAY ERR.])



(3) Pantalla de Configuración y configuración de puntos para modelo de Base de GX Developer



(a) Base model name

Designa el nombre del modelo de la unidad base instalada con 16 o menos caracteres. Modelo QCPU no utiliza el nombre del modelo.(Se utiliza como una memoria de usuario o para impresión de parametros)

(b) Power model name

Designa el nombre de modelo del módulo de alimentación instalado con 16 o menos caracteres. Modelo QCPU no utiliza el nombre del modelo. (Se utiliza como una memoria de usuario o para impresión de parametros)

(c) Extension Cable

Designa el nombre de modelo del cable de extensión en uso con 16 o menos caracteres. Modelo QCPU no utiliza el nombre del modelo . (Se utiliza como una memoria de usuario o para impresión de parametros)

(d) Slots (Utilizados con modelo QCPU)

Selecciona el número de puntos para el slot de la unidad base en uso de los siguientes:

- 2 (2 slots)
- 3 (3 slots)
- 5 (5 slots)
- 8 (8 slots)
- 10 (10 slots)
- 12 (12 slots)

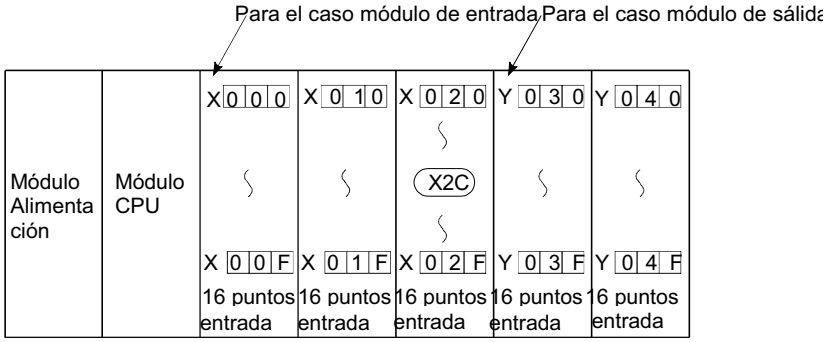
(e) 8 /12 Slots default(Utilizado con Modelo QCPU)

Selecciona cualquier opción para designar el número de slots para todas las unidades base al mismo número.

5.4 Cuales son los números E/ S?

Los Números de E/S son utilizados en programas de secuencias para importar datos ON/OFF para el modelo QCPU de la parte exterior y datos de salida ON/OFF desde el modelo QCPU hacia salida exterior.
La Entrada (X) es utilizada para la importación de datos ON/OFF para modelo QCPU.
La Salida (Y) es utilizada para la extracción de datos ON/OFF para el modelo QCPU.
Los números de E/S son expresados en hexadecimal.
Cuando utilizamos módulos E/S de 16-puntos, los números de E/S son asignados consecutivamente a los slots teniendo □□0 a □□F, 16 puntos, como sigue:
El módulo montado en la unidad base asigna lo siguiente:

- Para módulos de entrada, "X" se asigna en el comienzo del número de E/S.
- Para módulos de salida, "Y" se asigna en el comienzo del número de E/S.



5 - 7

5.5 Concepto de Asignación de Número E/S

5.5.1 Números E/S de la unidad base principal y la unidad base de extensión

La QCPU asigna el número de E/S en encendido o reset según los siguientes puntos. Como resultado, la QCPU puede ser controlada sin realizar la asignación de E/S mediante GX Developer.
Para asignar el número E/S , seguir los siguientes puntos:

(1) Número de slots de la unidad base

Los números de slots de la unidad base principal como de la base de extensión son programados en Base Mode. (Para Base mode, ver Sección 5.3.)

(a) En Auto mode, el número de slots se determina como el número de módulos permitidos montados en cada una de las unidades base. Por ejemplo, 5 slots asignados a una unidad base de 5-slot, y 12 slots asignados a una unidad base de 12-slot.

(b) En modo Detalle, el número de slots se determina como el designado en la pestaña “Asignación E/S” en el cuadro de dialogo “(PLC) Parameter".

(2) Orden de la asignación de E/S

Las E/S son asignadas en los módulos de izquierda a derecha consecutivamente, empezando por 0H asignado al módulo de la derecha de la QCPU de Altas Prestaciones de la unidad base principal.

(3) Orden de la asignación de E/S para unidad base de extensión

Las E/S para la unidad base de extensión continúan desde el último número asignado de E/S de la unidad base principal.
Las E/S son asignadas a la unidad base de extensión desde la izquierda (E/S 0) a la derecha consecutivamente, en el orden en que los conectores de la unidad base de extensión son puestos.

(4) E/S de cada slot

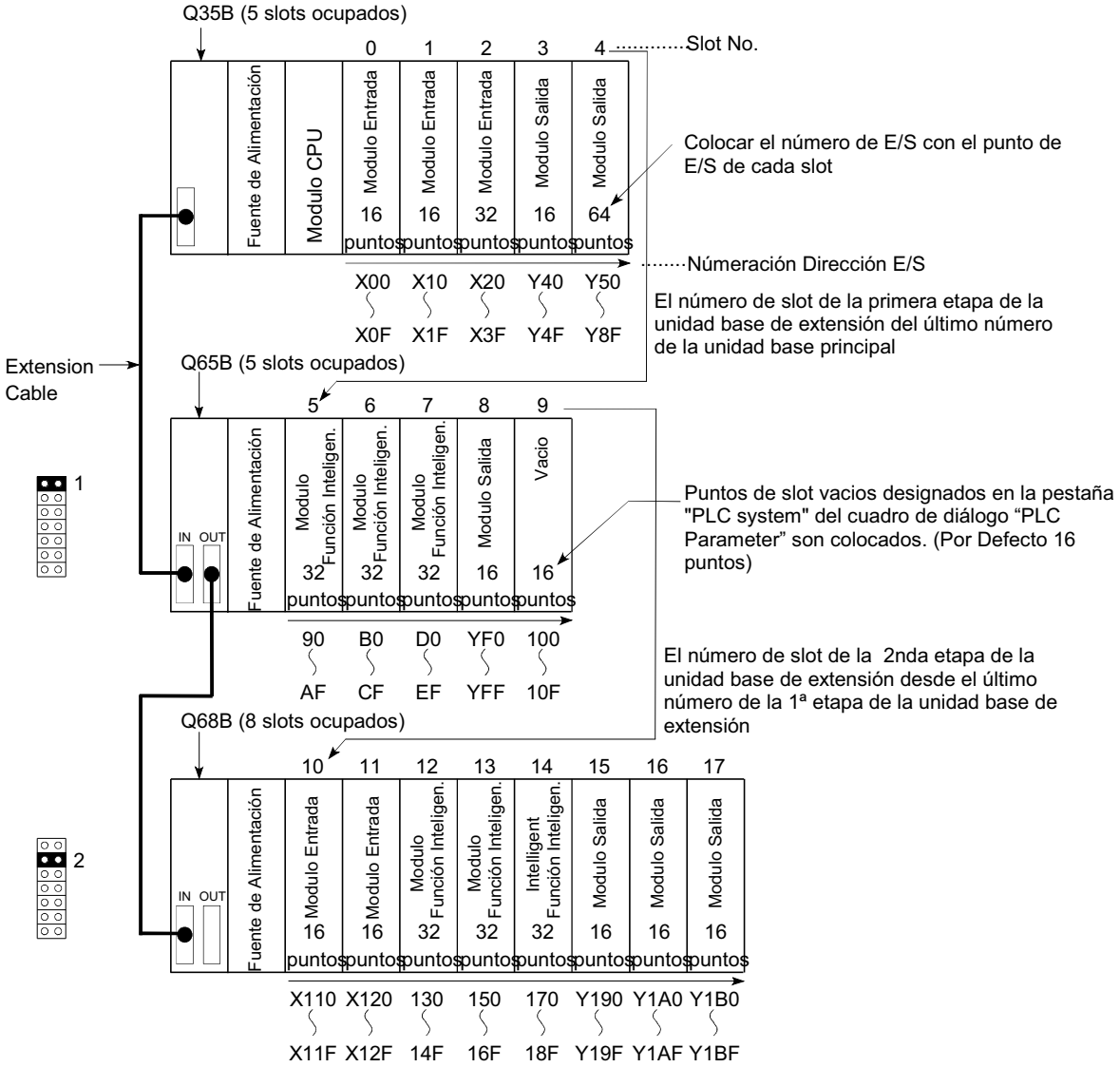
Cada slot de la unidad base ocupa los puntos de E/S de los módulos de E/S montados o módulos de función inteligente (módulos de función especial). Cuando un módulo de 32 puntos de entrada se monta en la parte derecha de una QCPU, X0 a X1F se asigna como números de E/S.

(5) E/S de slots vacíos

Si la unidad base tiene slots vacíos montados sin módulos de E/S o sin módulos de función inteligente (módulos de función especial), los puntos designados en la etiqueta “Asignación E/S” del cuadro de diálogo “PLC Parameter” son asignados a los slots vacíos.(Valor Por Defecto es 16 puntos.)

PUNTO
Cuando la asignación de la unidad base se realiza en modo Auto, el número de etapas vacías no se asegura incluso si la etapa de extensión es saltada en el número de estado del conector de la unidad base. (Se asignan primero los números de E/S bajos.) Para reservar etapas vacías para futuras expansiones, utilice el PLC Parameter configurando la unidad base.

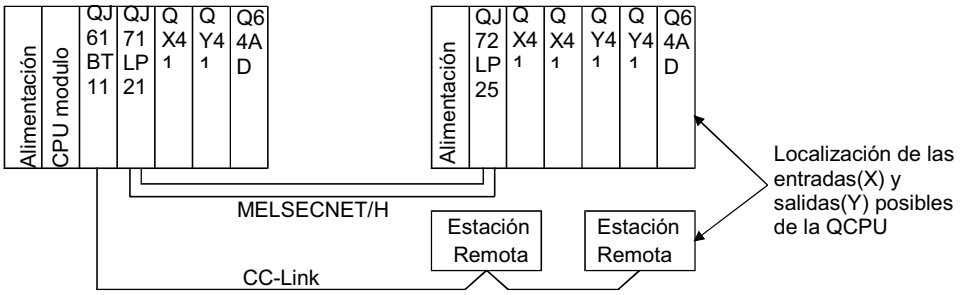
A continuación mostramos un ejemplo de asignación de E/S cuando la unidad base esta puesta en Auto mode sin asignación de E/S:


Q35B (5 slots ocupados)
0 1 2 3 4 ..... Slot No.
Fuente de Alimentación
Modulo CPU
Modulo Entrada
Modulo Entrada
Modulo Entrada
Modulo Salida
Modulo Salida
16 puntos
16 puntos
32 puntos
16 puntos
64 puntos
Colocar el número de E/S con el punto de E/S de cada slot
..... Númeración Dirección E/S
X00 X10 X20 Y40 Y50
X0F X1F X3F Y4F Y8F
El número de slot de la primera etapa de la unidad base de extensión del último número de la unidad base principal
Extension Cable
Q65B (5 slots ocupados)
5 6 7 8 9
Fuente de Alimentación
IN OUT
Modulo Función Inteligen.
Modulo Función Inteligen.
Modulo Función Inteligen.
Modulo Salida
Vacio
32 puntos
32 puntos
32 puntos
16 puntos
16 puntos
1
Puntos de slot vacios designados en la pestaña "PLC system" del cuadro de diálogo “PLC Parameter” son colocados. (Por Defecto 16 puntos)
90 B0 D0 YF0 100
AF CF EF YFF 10F
El número de slot de la 2nda etapa de la unidad base de extensión desde el último número de la 1ª etapa de la unidad base de extensión
Q68B (8 slots ocupados)
10 11 12 13 14 15 16 17
Fuente de Alimentación
IN OUT
Modulo Entrada
Modulo Entrada
Modulo Función Inteligen.
Modulo Función Inteligen.
Intelligent Función Inteligen.
Modulo Salida
Modulo Salida
Modulo Salida
16 puntos
16 puntos
32 puntos
32 puntos
32 puntos
16 puntos
16 puntos
16 puntos
2
X110 X120 130 150 170 Y190 Y1A0 Y1B0
X11F X12F 14F 16F 18F Y19F Y1AF Y1BF

PUNTO
El ejemplo anterior muestra el caso donde el módulo de función inteligente tiene 32 puntos de E/S. El número de puntos ocupados de E/S puede variar dependiendo del módulo de función inteligente. Remitirse al manual del módulo de función inteligente utilizado y controlar el número de E/S antes de la asignación de los números de E/S.

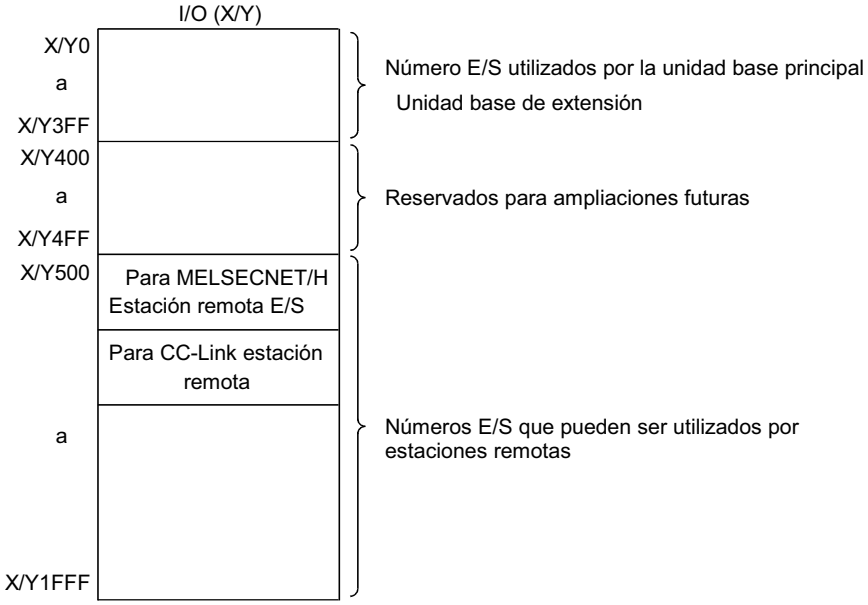
5.5.2 Estación Remota número E/S

Es posible adjudicar a una QCPU dispositivos de entrada(X) y salida(Y) de estaciones remotas con módulos de E/S, módulos con función inteligente y controlar los módulos en la red remota de MELSECNET/H, CC-Link y otros sistemas remotos de E/S.



Cuando se utiliza un dispositivo de una QCPU de entrada (X) y salida (Y) en estaciones remotas, serán asignados los números de E/S que siguen los números usados por la unidad base principal y unidades base de extensión.
Por ejemplo, si están siendo utilizados X/Y0 a X/Y3FF por la unidad base principal y unidad base de extensión, módulos de E/S y módulos de función inteligente, entonces los números posteriores a X/Y400 pueden ser utilizados por la estación remota.
No obstante, los números de E/S para estaciones remotas deberían tenerse en consideración además de la unidad base principal y unidades base de extensión, módulos de E/S y módulos de función inteligente.

Por ejemplo, si están siendo utilizados por la unidad base principal y unidades base de extensión 1024 puntos desde X/Y0 hasta X/Y3FF, y 256 puntos desde X/Y400 a X/Y4FF son para reservar para usos y adiciones futuras, entonces la situación mostrada en el diagrama siguiente.



PUNTO

Si la configuración de los parámetros de red no esta realizada en el sistema CC-Link, 2048 puntos en el rango desde X/Y1000 a X/Y17FF son asignadas al módulo master local de la CC-Link teniendo el número más bajo.

OBSERV.

Hay una restricción en el orden de colocación de los números de E/S para red MELSECNET/H remota de E/S, CC-Link y otras redes.

5.6 Asignación de E/S mediante GX Developer

Esta sección describe la asignación de E/S mediante GX Developer.

5.6.1 Determinación de E/S con

La Asignación de E/S se realiza con GX Developer bajo las siguientes circunstancias.

(1) Reservando puntos cuando se convierte a módulos distintos de 16-puntos
Se puede reservar el número de puntos previamente de forma que no tendrá que cambiar los números de E/S cuando el módulo actual sea cambiado por uno con diferentes números de E/S ocupados en el futuro.
Por ejemplo, usted puede asignar un módulo de 32-puntos E/S al slot donde un módulo de 16 puntos de E/S esté presente.

(2) Prevención de cambio de números de E/S cuando cambiamos módulos
Usted puede evitar el cambio en los números de E/S cuando un módulo de E/S diferente de un módulo de 16 puntos o un módulo de función inteligente (módulo función especial) es retirado a causa de un mal funcionamiento.

(3) Cambio de los números de E/S a los utilizados en programa
Cuando los números designados de E/S del programa son distintos de los números de E/S reales del sistema, cada módulo de E/S de la unidad base puede ser colocado en programa los números de E/S.

(4) Configuración del tiempo de respuesta de entrada de los módulos de entrada y módulos de interrupción (tiempo respuesta E/S)
Para hacer coincidir el tiempo de respuesta de entrada de los módulos de entrada y módulos de interrupción al sistema, seleccione "Type" en la pestaña "I/O assignment”.

(5) Configuración modo salida Error-tiempo
El modo salida en error puede ponerse en el módulo de salida, módulo E/S o módulos con función inteligente. Realice esta configuración después de seleccionar "Type" en la pestaña “I/O assignment”.

(6) Configuración de los módulos de función inteligente
Para configurar los módulos de función inteligente, seleccione "Type" en la pestaña "I/O assignment" .

(7) Configuración de salidas durante error de QCPU modelo de Altas Prestaciones
Para configurar el estado de las salidas (retener/borrar) de los módulos de salida y los módulos de función inteligente cuando el modelo de QCPU de Altas Prestaciones para la operación debido a un paro por error, seleccione “Type” en la pestaña "I/O assignment".

(8) Configuración Operación modelo QCPU de Altas Prestaciones durante un error de hardware de módulo con función inteligente
Para configurar QCPU modelo de Altas Prestaciones trabajando (continua/stop) durante un error de hardware de un módulo con función inteligente, seleccione "Type" en la pestaña "I/O assignment".

PUNTO

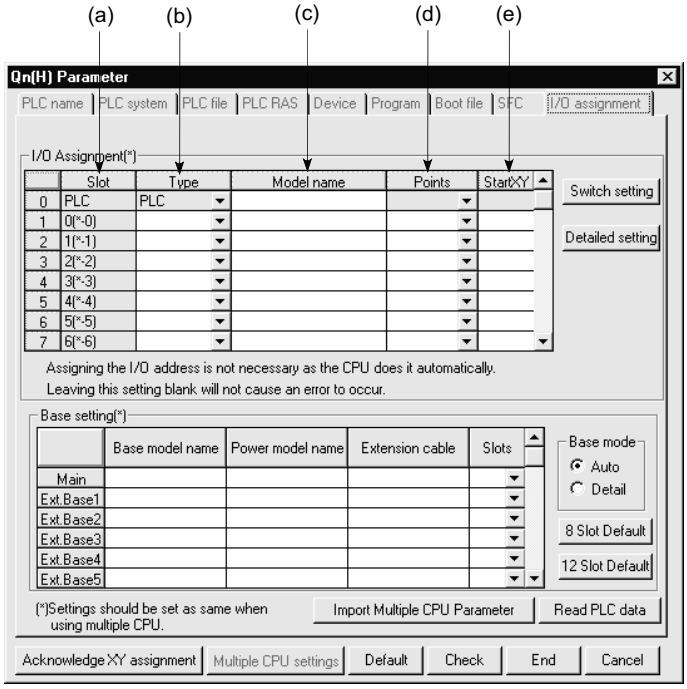
(1) La asignación de E/S es necesaria para cambiar el tiempo de respuesta de los módulos de entrada y de los módulos con función inteligente.
Para la asignación de E/S, configuración del tiempo de respuesta, configuración módulo con función inteligente y configuración salida modo error-tiempo, el PLC debe estar en OFF después a ON otra vez (ON a OFF a ON) o bien puede hacer reset de la QCPU.

(2) Si algún módulo de E/S diferente de módulos de 16-puntos falla sin haber realizado la asignación de E/S con GX Developer, los números de E/S posteriores a este módulo pueden cambiar, llevando a un mal funcionamiento. Por consiguiente se recomienda realizar la asignación de E/S mediante GX Developer.

5.6.2 Concepto de Asignación de E/S mediante GX Developer

(1) Asignación E/S para cada slot

"Type" (tipo módulos), "Points" (número de E/S ocupados), y "Start" (número de E/S) puede ser designado individualmente para cada slot de la unidad base.Por ejemplo, para cambiar el número de puntos de E/S ocupados de slots designado, sólo puede ser designado el número de puntos ocupados de E/S pueden ser designados.
El articulo diferente al designado es puesto en el estado donde la unidad base esta instalada.
La asignación de E/S se efectúa en a la pestaña "I/O assignment" en el cuadro de diálogo “(PLC) Parameter".



(a) Slot

Muestra el Nº de slot y la posición ordinal del slot en la unidad base.
Si la unidad base no es designada en modo “Detail”, el número de estado de la unidad base se muestra como " ＊ ", y el número ordinal del slot se cuenta desde el slot 0 de la unidad base principal.

(b) Type (Para modelo QCPU de Altas Prestaciones)

Seleccione el tipo de módulo que será montado de los siguientes:

- Empty (Slot vacio)
- Input (módulo de entrada)
- Hi Input (módulo de alta velocidad Serie Q)＊1
- Output (módulo salida)
- I/O Mix (módulo E/S mixto)
- Intelligent (módulo función inteligente o AnS correspondiente a un módulo de función especial)
- Interrupt (módulo interrupción Serie Q)＊2

Si el tipo no esta designado, se utiliza el tipo que este actualmente montado.

OBSERVACIÓN

＊1: "Hi input" puede ser configurada con GX Developer Versión 5 (productos posteriores a SW5D5C-GPPW-E)

*2: "Interrupt" puede ser configurada con GX Developer Versión 6 (productos posteriores a SW6D5C-GPPW-E)

(c) Model name
Designar el nombre de modelo del módulo montado con 16 caracteres o menos. Modelo QCPU de Altas Prestaciones no utiliza el nombre de modelo designado. (Se utiliza como apunte de usuario)

(d) Points (Utilizado con Modelo QCPU de Altas Prestaciones)
Para cambiar el número de los puntos ocupados de E/S de cada slot, seleccionelo de los siguientes:
- 0 (0 puntos)
- 16 (16 puntos)
- 32 (32 puntos)
- 48 (48 puntos)
- 64 (64 puntos)
- 128 (128 puntos)
- 256 (256 puntos)
- 512 (512 puntos)
- 1024 (1024 puntos)

Si el número de puntos de E/S ocupados no se designa para un slot, el módulo que esta montado actualmente será utilizado.

(e) Start XY (Utilizado con Modelo QCPU de Altas Prestaciones)
1) Cuando el número de E/S de cada slot es cambiado, usted deberia designar el número de cabecera de E/S de acuerdo con el cambio realizado.
Si Start XY no esta designado para un slot, el número de E/S continuando desde el último número del slot actual debe ser asignado.
2) Evitar la designación solapada de número de E/S de cada slot en los modelos de QCPU de Altas Prestaciones.
Da lugar a un error (SP. UNIT LAY ERR.) cuando se solapan los números de E/S .

(2) Estado del Slot después de la asignación de E/S

Cuando el número de E/S es asignado a un slot, la asignación del número de E/S tiene prioridad sobre la instalación actual del módulo.

(a) Si el número de puntos de E/S ocupados es menor que el módulo de E/S montado actualmente, algunos de los puntos de E/S del módulo no son utilizados.
Por ejemplo,si un slot donde hay un módulo de 32-puntos de entrada montado es designado para uno de 16-puntos de entrada, los últimos 16 puntos de los 32 puntos del módulo de entrada son deshabilitados.

(b) Si el número designado de puntos ocupados es menor que los que hay acutalmente montados del módulo de función inteligente ocurre un error, "SP. UNIT LAY ERR.".

(c) Si el número de puntos ocupados de E/S es mayor que los que estan montados actualmente en el módulo de E/S,los puntos en exceso de los del módulo actual son tomados como ficticios.

(d) Asegurese de poner el mismo tipo de módulo para el módulo montado y la asignación de E/S.
Si el tipo de módulo de la asignación de E/S es diferente que el que está montado actualmente en el sistema, este módulo puede que no esté trabajando correctamente.
Para los módulos de función inteligente, asegúrese que el número de puntos E/S son los mismos.

Módulo instalado Actual	Asignación E/S	Resultado
Módulo Entrada	Output/Vacio	Vacío
Módulo salida	Input/Vacio	Vacío
Módulo Entrada / módulo salida	Inteligente	Error (SP. UNIT LAY ERR.)
Módulo con función inteligente	Vacío	Vacío
	Entrada/Sálida	Error (SP. UNIT LAY ERR.)
Slot vacío	Inteligente	No ocurre error

(e) Asegurese de asignar los números de E/S de manera que el último número de E/S esté dentro del rango inferior a FFFH. Tiene lugar un error (SP. UNIT LAY ERR.) cuando el último número E/S excede de FFFH. (Sistema Monitor de GX Developer muestra"＊＊＊" como una dirección E/S.)

6 MEMORIAS Y ARCHIVOS TRATADOS POR EL MODELO QCPU DE ALTAS PRESTACIONES

(1) Archivos soportados por el modelo QCPU de altas prestaciones

(a) En el modelo QCPU de altas prestaciones los parametros, programas, comentarios, etc.se asignan "file names" y "extension", y son colocados en las siguientes memorias:

- Memoria ProgramaProgram memory
- Estándar ROM
- Tarjeta de Memoria

Cuando se lee y se escribe esta información desde GX Developer del modelo QCPU de altas prestaciones, los archivos pueden ser especificados por tipo (parametros, programa, comentarios, etc.) sin considerar su extensión. (GX Developer automáticamente asigna la extensión apropiada para el tipo de archivo que ha sido especificado.)

(b) Es imposible configurar y usar el mismo número de etapa de extensión con dos o más unidades de extensión base.

(2) Gestión de archivos con el modelo QCPU de altas prestaciones

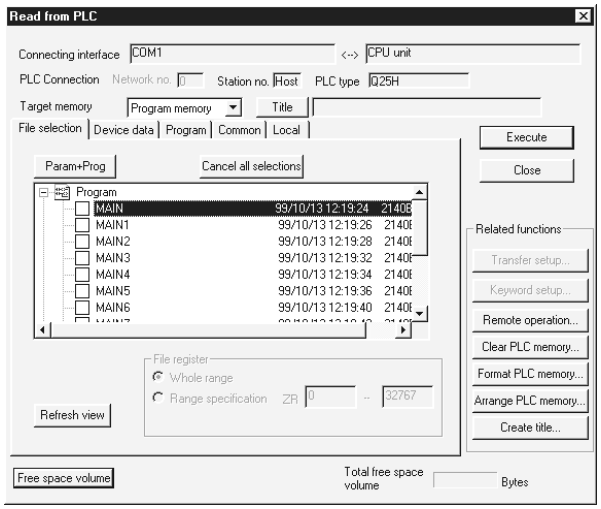
El uso de diferentes archivos y nombres de extensiones permite situar múltiples archivos en el modelo QCPU de altas prestaciones.
Como el modelo QCPU de altas prestaciones puede también incluso procesar un programa como si se tratara de un archivo, los programas creados pueden ser gestionados individualmente de acuerdo con su "diseñador", "proceso", o "función" utilizando diferentes nombres de archivo de programa.
Por otra parte, se permite la ejecución de un programa para ejecutar múltiples programas guardados en el modelo QCPU de altas prestaciones.
(Ver Capítulo 4 para detalles de modelo QCPU de altas prestaciones respecto detalles de ejecución de programa.)

(3) Escritura de Archivos desde GX Developer

El modelo QCPU de altas prestaciones guarda los archivos escritos desde GX Developer en la memoria (memoria programa/ ROM estándar /tarjeta de memoria).

(4) Detalles de Archivo

Nombre de archivo, tamaño y fecha de escritura, son establecido cuando se crea con GX Developer, son añadidas a todos los archivos escritos en modelos QCPU de altas prestaciones.
Cuando se monitoriza los archivos mediante GX Developer, los archivos son visualizados como se muestra.



(a) Nombre de archivo

1) El nombre de archivo consiste en el nombre de archivo (max. 8 caracteres) y la extensión (3 caracteres).
El nombre de un archivo escrito desde GX Developer en un modelo QCPU de altas prestaciones será visualizado en caracteres en mayúscula en pantalla. Cuando llame a un archivo utilizando una secuencia de programa, escriba el nombre de archivo en letras mayúsculas.
Una extensión que corresponde al tipo de archivo designado cuando el archivo fue escrito en el modelo QCPU de altas prestaciones mediante GX Developer es automaticamente agregada al nombre de archivo.

2) Las siguientes palabras reservadas por Windows no pueden ser utilizadas para un nombre de archivo.
• COM1 to COM9 • LPT1 to LPT9 • AUX • CON • PRN • NUL
• CLOCK$

(b) Fecha & hora
Se indica la fecha & hora cuando el archivo fue escrito desde GX Developer al modelo QCPU de altas prestaciones.
La fecha y tiempo configurado indica GX Developer-lateral fecha y tiempo.

(c) Tamaño
El tamaño del archivo cuando se escribe desde GX Developer al modelo QCPU de altas prestaciones es indicado en unidades de byte. (Para ver la última información del modelo QCPU de altas prestaciones, haga clic en el botón Update.)
Los Archivos son guardados en el modelo QCPU de altas prestaciones en archivo de programa y estándar ROM en unidades de 4-byte (1 paso), y en la tarjeta de memoria en unidades de 1-byte.
Cuando calculamos el tamaño de un archivo, al menos 64 bytes (136 bytes para programas) serán añadidos a todos los archivos de usuario creados diferentes de los archivos de registro.

6.1 Sobre la memoria del modelo QCPU de altas prestaciones

(1) Memoria de Usuario

Una memoria de usuario puede crearse con la memoria del modelo QCPU de altas prestaciones utilizando la secuencia de programa de GX Developer.
El modelo QCPU de altas prestaciones posee las siguientes memorias incorporadas:

- Memoria de Programa
- Estándar RAM
- Estándar ROM

Puede montar una tarjeta de memoria en el modelo QCPU de altas prestaciones para incrementar el tamaño de su memoria de usuario.

(a) Los programas utilizados para operaciones aritméticas con el modelo QCPU de altas prestaciones se guardan en la memoria de programa. Los programas guardados en la ROM estándar o en una tarjeta de memoria son leidos dentro de la memoria de programa para la operación aritmética.

(b) Los parametros y programas se guardan en la ROM estándar. Esta información se utiliza para las operaciones ROM del modelo QCPU de altas prestaciones.

(c) Registros de archivos y registros locales se guardan en la RAM estándar. El uso de registros de archivos en la RAM estándar permitirá alta velocidad de acceso como es el caso con los registros de datos.

(d) Tarjeta de Memoria. Puede conectar una tarjeta de memoria en la interfaz de tarjetas de memoria que posee el modelo QCPU de altas prestaciones. Esto permite la lectura y escritura de información. El modelo QCPU de altas prestaciones admite tres tipos de tarjeta de memoria: Tarjeta SRAM, Tarjeta Flash, y Tarjeta ATA .

1) La tarjeta SRAM permite la escritura y lectura de programas a través de una secuencia de programa en los siguientes casos donde:
   - Registros de archivos usados en exceso de la capacidad de la RAM estándar.
   - Se guarda el Muestreo de datos.
   - Se guarda el histórico de datos.

   El uso de archivos de registros permite la escritura y lectura de datos en 1017k puntos en una secuencia de programa.

2) La tarjeta Flash permite sólo la lectura a través de la secuencia de programa. La tarjeta Flash es muy útil cuando los datos escritos por el modelo QCPU de altas prestaciones son leidos a través de la secuencia de programa pero no se realiza ningún cambio en los datos.
   El uso de archivo de registros permite a una secuencia de programa la lectura de un máximo de 1018K puntos de los datos escritos por el modelo QCPU de altas prestaciones.

3) La tarjeta ATA se utiliza para informaición de usuario del PLC (datos de propósito general). Acceder a la información de usuario guardada en el PLC en la tarjeta ATA card puede hacerse con el formato CSV,

formato binario, utilizando una instrucción de acceso al registro en la secuencia de programa (e.g. FWRITE).

(2) Tipo de información almacenada en Memoria o Tarjeta de memoria del modelo QCPU de altas prestaciones

La tabla siguiente muestra el tipo de información almacenada en la RAM estándar/ ROM estándar o bien en la tarjeta de memoria.

Tipo información	Modelo QCPU de Altas Prestaciones Incorporada			Tarjeta de Memoria (RAM)	Tarjeta de Memoria (ROM)		Comentario
	Memoria de Programa	RAM Estándar	ROM Estándar	Tarjeta SRAM	Tarjeta Flash	Tarjeta ATA	
Parámetros	○	×	○	○	○	○	1 data/drive
Parámetros módulos de función inteligente	○	×	○	○	○	○	1 data/drive
Programa	◎	×	○ *1	○ *1	○ *1	○ *1	
Comentarios Dispositivos	○ *2	×	○	○	○	○ *3	
Valor inicial Dispositivo	○	×	○	○	○	○	
Archivos de registro	×	○ *5	×	○	○ *4	×	
Dispositivo local	×	○	×	○	×	×	1 data/CPU modulo
Datos depuración	×	×	×	○	×	×	
Fallo historia de datos	×	×	×	○	×	×	
PLC información de usuario	×	×	×	×	×	○ *6	

◎ : Necesaria, ○:Almacenada, × : No Almacenada

6 - 4

OBSERVACIONES

∗1: Cargar la memoria de programa para ejecutar un programa.
∗2: Los datos pueden ser escritos mediante GX Developer. Los comentarios no pueden ser utilizados en una instrucción de secuencia de programa.
∗3: La lectura de una secuencia de programa requiere varios scans.
∗4: Una secuencia de programa permite sólo la lectura. No se puede escribir ningún dato a través del acceso a una secuencia de programa.
∗5: Una RAM estándar mantiene un solo archivo.
∗6: Los datos pueden ser leídos o escritos con las siguientes instrucciones:
• S.FREAD (permite la lectura de un archivo especifico en la tarjeta de memoria)
• S.FWRITE (permite la escritura de un archivo especifico en la tarjeta de memoria)

La tabla siguiente muestra nombres de archivos y extensiones de datos de archivo guardados en el modelo QCPU de altas prestaciones o en una tarjeta de memoria.

Nombre Dato	Nombre Archivo
Parámetro	PARAM.QPA
Parámetro módulo con función inteligente	IPARAM.QPA
Programa	∗∗∗.QPG
Comentario Dispositivo	∗∗∗.QCD
Valor inicial Dispositivo	∗∗∗.QDI
Archivo de registro	∗∗∗.QDR
Dispositivo Local	∗∗∗.QDL
Información Depuración	∗∗∗.QTD
Histórico de dato de fallo	∗∗∗.QFD
Datos Usuario PLC	∗∗∗.∗∗∗

∗ Pueden ser nombradas por el usuario.

6 - 5

(3) Número de Dispositivo

(a) El modelo QCPU de Altas Prestaciones utiliza un número de dispositivo para controlar la RAM estándar, ROM estándar, y tarjetas de memoria. GX Developer especifica la memoria seleccionada (RAM estándar, ROM estándar o tarjeta de memoria) para leer/escribir parámetros y archivos de programa para el modelo QCPU de Altas Prestaciones. No existe la necesidad de especificar el número de dispositivo cuando estemos utilizando el GX Developer.

(b) La siguiente tabla muestra los números de dispositivo utilizados para especificar la memoria seleccionada (memoria de programa, RAM estándar, ROM estándar o tarjeta de memoria) cuando se utiliza una secuencia de programa. El número de dispositivo debe ser utilizado para especificar una memoria seleccionada cuando la lectura/escritura es realizada a través de un módulo serie de comunicación.

Memoria		Número de Dispositivo
Modelo QCPU de Altas Prestaciones incorporado	Memoria de Programa	0
	RAM estándar	3
	ROM estándar	4
Tarjeta de Memoria (RAM)	Tarjeta SRAM	1
Tarjeta de Memoria (ROM)	Tarjeta Flash	2
	Tarjeta ATA	2

(4) Capacidad y Formato Memoria

La siguiente tabla muestra el tamaño de la memoria del modelo QCPU de Altas Prestaciones y si es necesario o no el formatear la memoria.

		Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU	Formateado
Estándar RAM		64 kbytes	128 kbytes		256 kbytes		∗1
Memoria de Programa		28 k pasos (112 kbytes)	28 k pasos (112 kbytes)	60 k pasos (240 kbytes)	124 k pasos (496 kbytes)	252 k pasos (1008 kbytes)	∗1
Estándar ROM		112 kbytes	112 kbytes	240 kbytes	496 kbytes	1008 kbytes	∗2
Tarjeta de Memoria	Tarjeta SRAM	Q2MEM-1MBS: 1 Mbytes Q2MEM-2MBS: 2 Mbytes					Necesario. (Utilice GX Developer u ordenador personal)
	Tarjeta Flash	Q2MEM-2MBF: 2 Mbytes Q2MEM-4MBF: 4 Mbytes					No Necesario.
	Tarjeta ATA	Q2MEM-8MBA: 8 Mbytes Q2MEM-16MBA: 16 Mbytes Q2MEM-32MBA: 32 Mbytes					Necesario. (Utilice GX Developer u ordenador personal)

∗1: Si la memoria está en el estado inicial o está inestable por un voltaje bajo de la batería (Q6BAT), el formateado tiene lugar automáticamente cuando encendamos o reseteemos el PLC. Sin embargo, formatee con GX Developer antes de empezar el funcionamiento.

∗2: La ROM estándar se utiliza en la formación de la memoria de programa de la ROM, por lo tanto, es innecesario formatearla.

6.2 Memoria de Programa

(1) ¿Que es la Memoria de Programa?

(a) El modelo QCPU de altas prestaciones posee una memoria de programa que es una RAM interna que guarda programas ejecutados mediante el modelo QCPU de altas prestaciones.

(b) El almacenamiento de datos en la memoria de programa está soportada por baterías internas (Q6BAT) del modelo QCPU de altas prestaciones.

(c) Antes de utilizar el modelo QCPU de Altas Prestaciones por primera vez, la memoria de programa debe ser formateada por GX Developer.
Puede obtener más información sobre el procedimiento de formateo mediante GX Developer en los manuales de GX Developer.

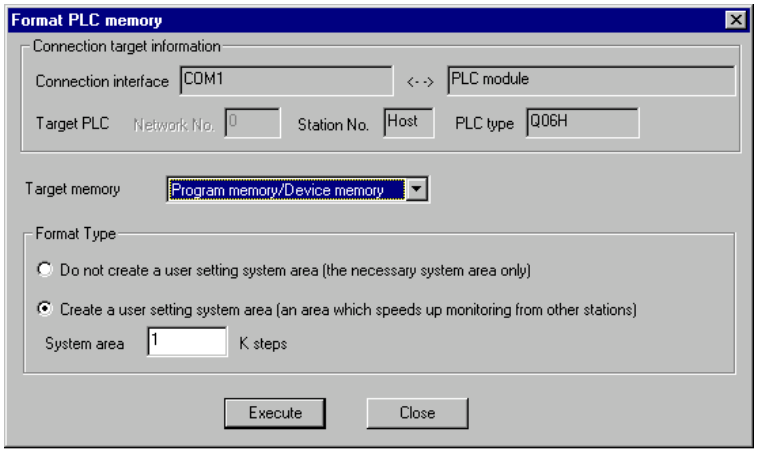
(2) Guardar Información

Datos de parámetros y programas pueden almacenarse en la memoria de programa. Para los tipos de información almacenados en la memoria de programa, ver sección 6.1.

(3) Formato

(a) Formatear

Escoja "Online" → "Format PLC memory" para abrir el cuadro de diálogo Formato Memoria PLC . Seleccione "Program memory/Device memory" de la lista de Target Memory.



(b) Capacidad de Memoria después de formatear

La capacidad de memoria de programa después de formatearse es como sigue.

Tabla 6.1 Capacidad de Memoria después de formatear ∗1

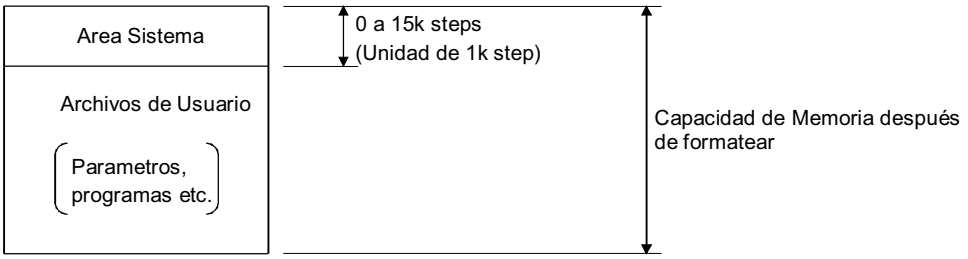
Nombre modelo CPU	Memoria ∗2	Max. Número de archivos guardados
Q02CPU	28 k pasos (114688 bytes)	28 archivos
Q02HCPU	28 k pasos (114688 bytes)	28 archivos
Q06HCPU	60 k pasos (245760 bytes)	60 archivos
Q12HCPU	124 k pasos (507904 bytes)	124 archivos
Q25HCPU	252 k pasos (1032192 bytes)	252 archivos ∗3

(c) Precauciones para el formateado

1) Formateado de la memoria de programa

La memoria de programa del modelo QCPU de Altas Prestaciones puede ser solamente utilizada después de ser formateada mediante GX Developer.

Cuando formateamos la memoria de programa, designar o no si un area del sistema sirve para asignar ajustes de usuario o cambios de múltiples bloques online.(0 a 15 k pasos (en 1 k-paso unidad) puede ser asignada para los ajustes de usuario del area de sistema.)



2) Configuración area Sistema

Si las interfaces RS-232 y USB están conectadas a GX Developer, el area de sistema de información de usuario será utilizada para registrar información de monitorización (monitor data) de GX Developer conectado al módulo de comunicación serie.

La asignación de espacio para el sistema y areas de usuario definidas hará mucho más fácil trabajar con la función de monitorización mediante GX Developer conectado al módulo serie de comunicación.

Aunque la designación del area de ajustes de usuario acelerará la monitorización de GX Developer conectado al módulo de comunicación serie, incluso reduce la cantidad de espacio disponible para los archivos de usuario.

Observaciones

1)∗1: Esta tabla muestra un ejemplo dónde 0 k paso se asigna para un área de sistema.

2)∗2: En computación de la capacidad de memoria, 1 paso = 4 bytes.

3)∗3: Nº Máximo de programas ejecutables es 124. Más de 124 no es posible

6.3 Acerca de la ROM estándar

(1) ¿Que es la ROM estándar?

(a) La ROM estándar es utilizada para el funcionamiento de la ROM del modelo QCPU de Altas Prestaciones.

(b) Los programas almacenados en la ROM estándar pueden ser usados después de ser leída a la memoria de programa en concordancia con la configuración realizada en la pestaña "Boot file" en el cuadro de diálogo “(PLC) Parameter" .

(c) La ROM estándar no necesita ser formateada.

(d) Para escribir en la ROM estándar se efectúa con GX Developer en On-line "PLC Writing" (flash ROM) de "Create Program Memory ROM" (ver Sección 6.6.1.) También es posible escribir en la ROM estándar de la tarjeta de memoria con "Automatic writing in the estándar ROM" sin utilizar GX Developer.

PUNTOS

(1) Antes de escribir información en la ROM estándar, toda la información previa guardada en la ROM estándar será borrada. Por lo tanto, toda la información guardada en la ROM estándar debe ser leída y copiada en la memoria de programa primeramente. Después, leerla y modificarla como sea necesario. Finalmente, al mismo tiempo escriba la información modificada en la ROM estándar. Por favor observe que puede tener lugar un error si la información guardada en la ROM estándar es utilizada en una secuencia de programa, con información escribiéndose en la ROM estándar.

(2) Para más detalles en el procedimiento de formateado mediante GX Developer, referirse a los manuales de GX Developer.

(2) Guardando información

Una ROM estándar guarda información, tanto parámetros como programas. Ver sección 6.1 para guardar información en la ROM estándar.

(3) Capacidad de Memoria

Tabla 6.2 muestra la capacidad de memoria de la estándar ROM.

Tabla 6.2 Capacidad Memoria

Nombre de Modelo	Capacidad de Memoria *2	Máx. Número de archivos guardados
Q02CPU	28 k pasos (114688 bytes)	28 archivos
Q02HCPU	28 k pasos (114688 bytes)	28 archivos
Q06HCPU	60 k pasos (245760 bytes)	60 archivos
Q12HCPU	124 k pasos (507904 bytes)	124 archivos
Q25HCPU	252 k pasos (1032192 bytes)	252 archivos

OBSERVACIONES

*2 En computación de la capacidad de memoria, 1 paso = 4 bytes.

6.4 Acerca de la RAM Estándar

(1) Que es la RAM estándar

(a) La RAM estándar es utilizada cuando se usan registros de archivo o dispositivos locales sin tarjeta de memoria montada en el modelo QCPU de Altas Prestaciones.

(b) La RAM estándar debe ser formateada con GX Developer cuando se utiliza el modelo QCPU de Altas Prestaciones por primera vez. Referirse al manual de GX Developer para ver más detalles del método de formateado.

(c) La Información puede ser escrita en la RAM estándar utilizando la función online: "Write to PLC."

(2) Información Guardada

La RAM estándar mantiene dos archivos: registro de archivos y archivo dispositivos locales. Cualquier otro archivo no puede ser escrito en la RAM estándar.
(La RAM estándar no puede guardar otros archivos que los registros de archivo y archivo de dispositivos locales.)

(3) Formateado

(a) Formateado
Para formatear una RAM estándar, escoja "Online" → "Format PLC memory" y seleccione "Estándar RAM" en el listado "Target memory" .Ver Sección 6.2 para el cuadro de diálogo “PLC Memory Format”.

(b) Capacidad de Memoria después de formatear
Tabla 6.3 muestra la capacidad de memoria de la RAM estándar.

Tabla 6.3 Capacidad de Memoria

Tipo CPU	Nº Serie	Número de archivos guardados	Número de archivos guardados	
			Registro de Archivos	Dispositivo Local
Q02CPU	—	32 k palabras (64 kbytes)	1	1
Q02HCPU	"04011" o anterior	32 k palabras (64 kbytes)	1	1
	"04012" o posterior	64 k palabras (128 kbytes)		
Q06HCPU	"04011" o anterior	32 k palabras (64 kbytes)	1	1
	"04012" o posterior	64 k palabras (128 kbytes)		
Q12HCPU	"02091" o anterior	32 k palabras (64 kbytes)	1	1
	"02092" o posterior	128 k palabras (256 kbytes)		
25HCPU	"02091" o anterior	32 k palabras (64 kbytes)	1	1
	"02092" o posterior	128 k palabras (256 kbytes)		

(4) Precauciones

Cuando configuramos archivos de registros y dispositivos locales en la RAM estándar, la capacidad de memoria es asegurada en unidades de 1024 byte para Q12HCPU/Q25HCPU que tienen el número de serie cuyos 5 dígitos más altos son “02092” o posteriores.
Capacidad de memoria es asegurada en unidades de 512 byte para Q12HCPU/Q25HCPU, Q02CPU, Q02HCPU y Q06HCPU que tienen el número de serie cuyos 5 dígitos superiores son "02091" o anteriores.

PUNTO

Cuando especificamos registros de archivo utilizando el método de acceso serie (ZR□) con los comandos que acceden a la estándar RAM en "02092" y posterior del modelo de Altas Prestaciones QCPU, el tiempo necesario para el procesado de cada comando de "02091" y modelos de Altas Prestaciones QCPU previos será prolongado. (QnCPU: Average 0.65μs, QnHCPU: Average 1.1μs)
Mostramos en la tabla siguiente el tiempo de procesamiento cuando se utiliza el comando MOV.

Comando	Q12HCPU		Q02CPU	
	“02092” o posterior	“02091” o anterior	“02092” o posterior	“02091” o anterior
MOV K0 R0	0.11	0.11	0.26	0.26
MOV K0 ZR0	3.55	2.88	7.71	6.64

6.5 Tarjeta de Memoria

(1) Tarjeta de Memoria

(a) Una tarjeta de memoria se utiliza para aumentar el tamaño de la memoria interna del modelo QCPU de Altas Prestaciones.

(b) Hay tres tipos de tarjetas de memoria para utilizar en el modelo de Altas Prestaciones QCPU: tarjeta SRAM ,tarjeta Flash, y tarjeta ATA .

PUNTOS

(1) Antes de utilizar la tarjeta de memoria por primera vez, la tarjeta de memoria debe ser formateado mediante GX Developer.
Para más detalles para el procedimiento de formateo mediante GX Developer, referirse a los manuales de GX Developer.

(2) Antes de escribir información en la tarjeta Flash, toda la información previa guardada en la tarjeta Flash será borrada. Por este motivo, para escribir información en la tarjeta Flash usted debe primero leer y copiar toda la información previa guardada en la tarjeta Flash antes de escribir información nueva. Por favor vea que puede tener lugar un error si la información guardada en la tarjeta Flash es utilizada en un programa de secuencia, con información siendo escrita en la tarjeta Flash.

(2) Información almacenada

Una tarjeta de memoria mantiene tanto información de parámetros como de programa. Ver Sección 6.1 para los tipos de información guardad en la tarjeta de memoria.

(3) Formato

La tarjeta SRAM y ATA debe ser todas formateadas.
Hasta que la tarjeta SRAM y la tarjeta ATA adquiridas no esten todavía formateadas, utilícelas después de formatear con GX Developer.
(La Tarjeta Flash no requiere ser formateada.)

(a) Ejecución de formateo
Para dar formato a la tarjeta, escoja "Online" _ "Format PLC memory" y entonces seleccione "Memory card (RAM)" o "Memory card (ROM)" en GX Developer.
(Referirse a la Sección 6.2 para la pantalla Format PLC memory .)

(b) Precauciones
Cuando la tarjeta SRAM o tarjeta ATA es formateada, la "memory card information area" es automáticamente asegurada, reduciendo el espacio mediante el tamaño de "memory card information area".

(4) Capacidad de Memoria

Tabla 6.4 indica la capacidad de memoria de las tarjetas de memoria.

Tabla 6.4 Capacidad de Memoria

Tipo	Nombre Modelo Tarjeta de Memoria	Capacidad de Memoria	Número de archivos guardados
Tarjeta SRAM	Q2MEM-1MBS	1011.5 kbytes ＊1	256 archivos
	Q2MEM-2MBS	2034 kbytes ＊1	288 archivos
Tarjeta Flash	Q2MEM-2MBF	2035 kbytes	288 archivos
	Q2MEM-4MBF	4079 evite	288 archivos
Tarjeta ATA	Q2MEM-8MBA	7940 kbytes ＊1	512 archivos
	Q2MEM-16MBA	15932 evite ＊1	512 archivos
	Q2MEM-32MBA	31854 evite ＊1	512 archivos

＊1: Las capacidades indicadas de la tarjeta SRAM y ATA son después de ser formateadas.

6.6 Escritura en la ROM estándar o la tarjeta Flash

6.6.1 Escritura en la ROM estándar o la tarjeta Flash mediante GX Developer

La función "write to PLC" en menú Online de GX Developer no permite al usuario escribir archivos en la ROM estándar o en la tarjeta Flash. Para escribir archivos en la ROM estándar o la tarjeta Flash trabajando desde GX Developer, el menú Online de GX Developer contiene dos funciones: "Write the program memory to ROM" y "Write to PLC (Flash ROM)."

(1) Escritura de la memoria de programa en ROM

(a) La función "Write the program memory to ROM" permite a un conjunto de archivos guardados en la memoria de programa ser escritos en la ROM estándar o tarjeta Flash. Está función es utilizada para depurar los programas guardados en la memoria de programa.

(b) Cuando se ejecuta la función "Write a memory to ROM" , todos los archivos guardados en la ROM estándar o tarjeta Flash son borrados antes de que sean escritos el conjunto de archivos guardados en la memoria de programa. No puede añadirse ningún archivo a la ROM estándar o la tarjeta Flash.

(c) La capacidad de memoria de la ROM stándar o la tarjeta Flash es igual que el de la memoria de programa. Una memoria de tamaño mayor que la capacidad de memoria de la memoria de programa no puede ser utilizada.

(d) Para ejecutar la función "Write the program memory to ROM" , configure la longitud del time-check de GX developer en 60 segundos o más. Un time-check menor puede causar en GX Developer un time-out. Para ejecutar la función "Write the program memory to ROM" via CC-Link network trabajando desde GX Developer en una estación local, configure la longitud del tiempo de CPU de monitoreo CC-Link (SW0A) en 60 segundos o más. Por defecto és 90 segundos. Utilice el valor por defecto cuando realice la configuración.

(2) Escritura en PLC (Flash ROM)

(a) La función "Write to PLC (Flash ROM)" es utilizada para escribir un grupo de archivos especificados por GX Developer a ROM estándar o tarjeta Flash.

(b) La función "Write to PLC (Flash ROM)" puede rellenar todo el espacio disponible en la ROM estándar o la tarjeta Flash. Si un programa que contiene un número pequeño de pasos se escribe en la tarjeta Flash, tardará mucho en rellenar la tarjeta Flash con programas. Cuando la interfaz RS-232 se monta Q2MEM-4MBF, un baud rate de 115.2k bps necesita aproximadamente 14 minutos. Para escribir información en la tarjeta Flash, incremente el baud rate o bien utilice la interfaz USB. Si la función "Write to PLC (Flash ROM)" se ejecuta desde una estacicón local, el tiempo de comuncicación será superior.

(c) Para ejecutar la función "Write to PLC (Flash ROM)" , configure la longitud del time-check en 60 segundos o más en GX Developer. Un time-check pequeño puede causar time-out en GX Developer. Para ejecutar la función "Write to PLC (Flash ROM)" via CC-Link network desde GX Developer en

una estación local, configure la longitud del tiempo de CPU de monitoreo CC-Link (SW0A) en 60 segundos o más. Por defecto és 90 segundos. Utilice el valor por defecto cuando realice la configuración.

(d) Cuando la función "Write to PLC (Flash ROM)" es ejecutada, todos los archivos guardados en la ROM estándar o en la tarjeta Flash son borrados antes de que se escriban los archivos especificados mediante GX Developer. No puede añadirse ningún archivo a la ROM estándar o la tarjeta Flash. Para añadir nuevos archivos a archivos antiguos, lea todos los archivos antiguos del modelo QCPU de Altas Prestaciones y escribalos otra vez en el modelo QCPU de Altas Prestaciones.

(e) La función "Write to PLC (Flash ROM)" puede ser ejecutada cuando el modelo QCPU de Altas Prestaciones esta en estado RUN. Sin embargo, en los siguientes casos, ejecute la función "Write to PLC (Flash ROM)" después que el modelo QCPU de Altas Prestaciones entre en estado de STOP.

1) El archivo de registros de la tarjeta Flash es utilizado en la secuencia de programa.

2) Los archivos de registros son utilizados en una secuencia de programa configurando el archivo de registro como "set not to use" en “PLC parameter”.

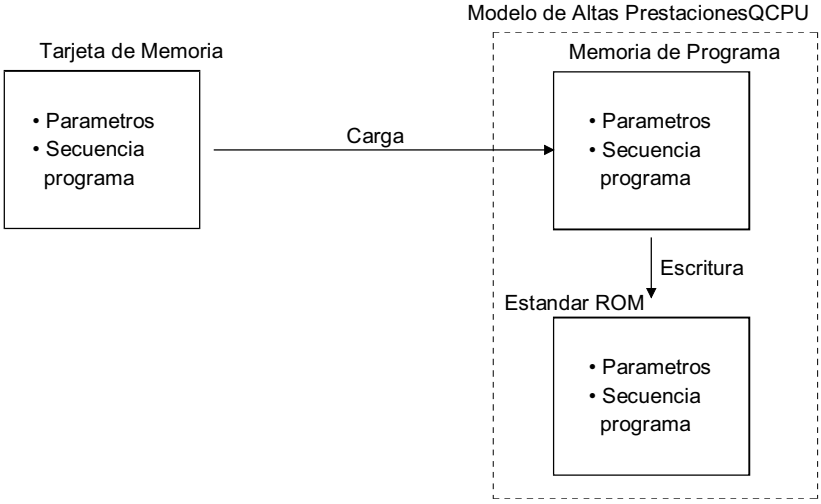
Si la función "Write to PLC (Flash ROM)" se ejecuta cuando el modelo QCPU de Altas Prestaciones esta en estado RUN, puede tener lugar un error en el modelo QCPU de Altas Prestaciones, este puede dejar de funcionar.

(f) Mientras la función "Write to PLC (Flash ROM)" se ejecuta, la lectura / escritura no puede realizarse desde otros módulos. Esto podría causar time out en los otros módulos.

PUNTO
Cuando el modelo QCPU de Altas Prestaciones es puesto al estado STOP y esta realizando “Write to PLC” (Flash ROM), no poner en estado RUN. RUN no puede realizarse con normalidad durante” Write to PLC“(Flash ROM). Ejecute RUN después de que”Write to PLC” (Flash ROM) esté completado.

6.6.2 Escritura Automática en la ROM estándar (Auto descarga de toda la información de la tarjeta de Memoria a la ROM estándar)

La función "Automatic write to estándar ROM" escribe los parámetros y la secuencia de programa guardada en la tarjeta de memoria en la ROM estándar del modelo QCPU de Altas Prestaciones sin utilizar GX Developer. (La escritura de los parámetros y la secuencia de programa en la tarjeta de memoria es ejecutada mediante GX Developer (Versión 6 o posterior.)
Con esta función, los parámetros y la secuencia de programa son cargados de la tarjeta de memoria a la memoria de programa, y los parámetros cargados y secuencia de programa son entonces escritos desde la memoria de programa en la ROM estándar tal y como se muestra en la imagen siguiente.



Se utiliza "Automatic write to standard ROM" para cambiar los programas del modelo QCPU de Altas Prestaciones que trabajan con operaciones ROM estándar. Sobrescribir en la ROM estándar se realiza mediante GX Developer, pero utilizando "Automatic write to standard ROM" mueve la tarjeta de memoria en la cual los parámetros y los programas cambiados son escritos al modelo QCPU de Altas Prestaciones, de tal manera que son escritos en la ROM estándar desde la tarjeta de memoria.

"Automatic write to standard ROM" es compatible con la combinación del modelo QCPU de Altas Prestaciones con los cinco dígitos de su número de serie, como "02092" o posteriores y Versión 6 o posterior de GX Developer .
Si la tarjeta de memoria, donde "Automatic write to standard ROM" esta, montada en el modelo de Altas Prestaciones QCPU cuyos cinco dígitos más altos de su número de serie es "02092," o posterior las operaciones de carga de programa son realizadas desde la estándar ROM.

Requerimientos para "Automatic write to standard ROM".
- Poner "Automatic write to standard ROM settings" en “(PLC) Parameters dialog box".
- Tarjeta de Memoria donde los parámetros y programas están guardados.
- Tarjeta de Memoria montada sobre el modelo de Altas Prestaciones QCPU y los cambios de configuración del modelo de Altas Prestaciones QCPU.

PUNTO

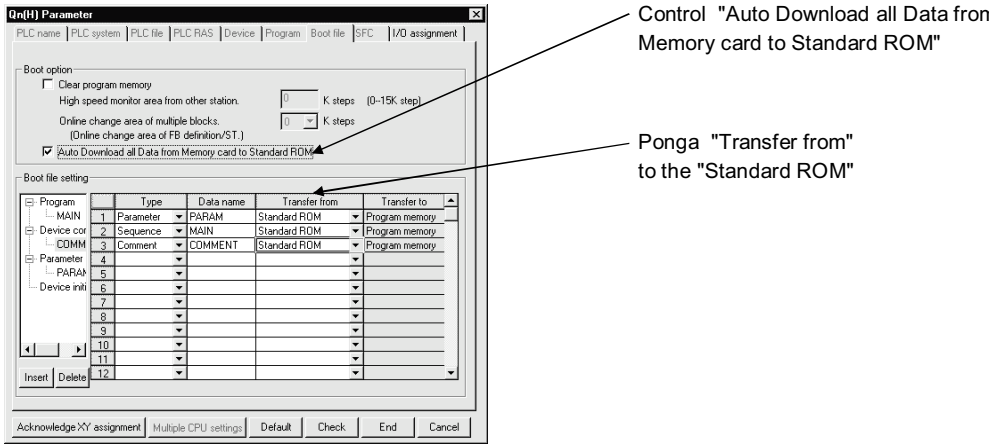
Realizar "Automatic write to standard ROM" después de que el control del modelo QCPU de Altas Prestaciones se ha suspendido.
Un error de suspensión (BOOT OK (Error Code: 9020)) ocurre cuando se completa una escritura automática sobre la ROM estándar.
Es necesario realizar un reset al modelo de Altas Prestaciones QCPU o bien, reiniciar la alimentación del PLC después de completar"Automatic write to standard ROM" .

(1) Procedimiento de Ejecución para "Automatic write to standard ROM"

Observe el siguiente procedimiento para "Automatic write to standard ROM".

(a) Funcionamiento con GX Developer (Configuración para "Automatic write to standard ROM")

1) Control "Auto Download all Data from Memory card to standard ROM" en la pestaña "Boot file" en el cuadro de diálogo “(PLC) Parameter". Coloque los parametros y programas que serán cargados. (Ponga "Transfer from" a "Estándar ROM".)



2) Guarde los parámetros de instalación y los programas que serán cargados en la tarjeta de Memoria.

(b) Funcionamiento del modelo QCPU de Altas Prestaciones (Escritura Automatica en la estándar ROM)

1) Apague la alimentación del PLC.
2) Monte la tarjeta de memoria que contiene los parametros y programas que serán cargados sobre el modelo de Altas Prestaciones QCPU.
3) Ponga los parámetros válidos de dispositivo a la tarjeta de memoria mediante los microruptores siguientes:
   • Cuando se monta una tarjeta SRAM : SW2 : ON, SW3 : OFF
   • Cuando se monta una tarjeta Flash/ATA : SW2 : OFF, SW3 : ON
4) Active la alimentación del PLC.
   Cargue el archivo especifico con los parámetros de la memoria de programa de la tarjeta de memoria. Escriba los contenidos de la memoria de programa en la ROM estándar cuando este cargado por completo.
5) El LED "BOOT" parpadeará cuando la escritura automática sobre la ROM estándar este completada, y el modelo QCPU de Altas Prestaciones asumirá el estado de suspensión error.
6) Apague la alimentación del PLC.
7) Extraiga la tarjeta de memoria, y después ponga los parámetros válidos en la ROM estándar con los microruptores de la CPU:
   • Estándar ROM: SW2 : ON, SW3 : ON

(c) Los parámetros y programas serán cargados de la ROM estándar a la memoria de programa para habilitar operaciones actuales cuando el PLC esta en ON.

(2) Precauciones

Esta sección indica las precauciones para realizar "Automatic write to estándar ROM"

(a) Si el archivo que ha de cargarse desde la tarjeta de memoria comparte el mismo nombre que el archivo de la memoria de programa, la tarjeta de memoria será sobreescrita. También, si el archivo que será cargado de la tarjeta de memoria no comparte el mismo nombre que el archivo en la memoria de programa, este será añadido a la memoria de programa. El error "FILE SET ERROR (Error code: 2401)" ocurrirá en ese momento si la capacidad de la memoria de programa se excede.

(b) Es posible seleccionar como realizar la carga después de haber borrado la memoria de programa, o bien realizar la carga sin realizar el borrado de la memoria de programa cuando cargamos de la tarjeta de memoria a la memoria de programa.
Realizar el arranque después de haber borrado la memoria de programa cuando "Automatic write to estándar ROM" evita el del desbordamiento de la memoria de programa durante la carga.

(c) La configuración "Auto Download all Data from Memory card to Estándar ROM" en la pestaña "Boot file" es valida sólo cuando los parámetros del modelo QCPU de Altas Prestaciones es para "Memory Card".
La configuración "Auto Download all Data from Memory card to standard ROM" en la pestaña "Boot file" esta desactivada si los parámetros válidos están puestos en "Program Memory" o "Estándar ROM".

6.7 Ejecución de programas de ROM estándar de la tarjeta de memoria(ejecución de arranque)

(1) Ejecución programas modelo QCPU de Altas Prestaciones

(a) El modelo QCPU de Altas Prestaciones ejecuta programas guardados en la memoria de programa.
El modelo QCPU de Altas Prestaciones no realiza operaciones de programas guardados en la ROM estándar o tarjeta de memoria.

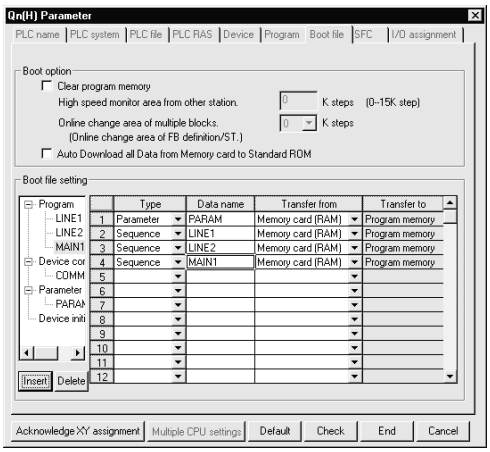
(b) Para ejecutar programas guardados en la ROM estándar y la tarjeta de memoria, designar nombres de archivo de arranque (leer) en la memoria de programa en la pestaña "Boot file" en el cuadro de diálogo “(PLC) Parameter".
Programas con nombre de archivo designado se cargan desde la ROM estándar/ tarjeta de memoria a la memoria de programa y ejecución cuando la alimentación pasa a ON o bien cuando se hace un reset del modelo QCPU de Altas Prestaciones.

(2) Preparación para ejecución de arranque.

Realice los siguientes pasos para la preparación de la ejecución de arranque:

(a) Cree un programa utilizando GX Developer.
Cree un programa utilizado para la ejecución de arranque.

(b) Seleccione un archivo a cargar mediante GX Developer.
Seleccione el archivo de carga en la pestaña "Boot file" del cuadro de diálogo “(PLC) Parameter".



(c) Realice la configuración del Hardware del modelo QCPU de Altas Prestaciones. Ponga los selectores Dip para especificar los parametros a comandar del dispositivo del modelo QCPU e Altas Prestaciones.

(d) Inserte una tarjeta de memoria.
Inserte una tarjeta de memoria en el slot si usted quiere guardar los parámetros o los programas en la tarjeta de memoria durante la ejecución de arranque.

(e) Escriba parámetros y programas utilizando GX Developer.
Escriba los parámetros en el dispositivo de parámetros. Escriba el programa en la memoria especificada en la pestaña "Boot file" en el cuadro de diálogo “(PLC) Parameter".

(f) Ejecutar el programa.
Resetea el modelo QCPU de Altas Prestaciones con el microruptor

RESET/L.CLR . Después de la carga esta completada en la memoria especificada, el LED "BOOT" se iluminará.

(3) Cambio archivos de programa mientras el modelo QCPU de Altas Prestaciones esta en estado Run.

(a) Mientras el modelo QCPU de Altas Prestaciones esta en estado RUN, añadir/cambiar/borrar de los archivos de programa de la ROM estándar o de la tarjeta de memoria a la memoria de programa puede ser realizado utilizando cualquier instrucción de las siguientes en la secuencia de programa.
• PLOAD (Cargando el programa de la tarjeta de memoria)
• PUNLAOD (Descarga programa de la memoria de programa)
• PSWAP (Carga + Descarga)
Para tener más detalles de las instrucciones PLOAD, PUNLAOD y PSWAP, referirse al Programming Manual (Common Instructions) de QCPU (Q Mode)/QnACPU.

(b) Incluso si un archivo de programa se cambia mientras el modelo QCPU de Altas Prestaciones esta en estado RUN, la configuración especificada en la pestaña "Program" del cuadro de diálogo “(PLC) Parameter" permanecerá inalterada. Cuando el modelo QCPU de Altas Prestaciones está en estado STOP, las configuraciones realizadas en la pestaña "Program" en el cuadro de diálogo “(PLC) Parameter" debe ser ajustada a cualquier cambio (adición, cambio o borrado de los nombres de programa) realizado cuando el modelo QCPU de Altas Prestaciones estaba en estado RUN.
Si no se ha realizado ningún ajuste en la pestaña "Program" en el cuadro de diálogo “(PLC) Parameter", puede tener lugar un error cuando el modelo QCPU de Altas Prestaciones entra en estado RUN desde el estado STOP.

(4) Precauciones en la Ejecución de Programas en la ROM Estándar/ Tarjeta de Memoria

(a) Para la ejecución de arranque, guarde los parámetros (PLC parameters) del archivo de configuración de carga en la RAM estándar o bien la tarjeta de memoria. Si los parámetros son guardados en la memoria de programa y un dispositivo con los parámetros válidos puesto en "Program Memory", la configuración del archivo a cargar realizado en el cuadro de diálogo “(PLC) Parameter" es ignorado. Como resultado, la ejecución de arranque no se realiza cuando está activada la alimentación o bien cuando se realiza un reset del PLC.

(b) Si se escriben programas en la memoria de programa durante el estado RUN mientras se esta realizando la carga de un programa mediante una tarjeta de memoria (RAM), cualquier cambio realizado será reflejado en los programas guardados en la tarjeta de memoria (RAM). Para más detalles respecto de la escritura de programas durante el estado RUN, ver sección 7.10.

(c) Si se escriben programas en la memoria de programa durante el estado RUN mientras realizamos una carga de programa desde la ROM estándar/ tarjeta de memoria(ROM), ningún cambio realizado no será reflejado en los programas guardados en la estándar ROM o en la tarjeta de memoria (ROM).

(d) En la pestaña "Boot file" del cuadro de diálogo “(PLC) Parameter", ponga el máximo número de archivos de carga a los números de archivo guardados en la memoria de programa. El número de archivos de arranque será decrementado por alguno de estos motivos donde:
• Se especifica un arranque .
• Un parámetro del PLC donde se ha configurado el archivo de arranque y se ha cargado.

(e) Si la operación de arranque se realiza bajo las siguientes condiciones, tardará un máximo de 200 ms para cada 1k pasos (4kbyte) durante la secuencia de arranque.
• Para arranque desde la tarjeta ATA card.
• Para arranque desde la ROM estándar con una tarjeta ATA card montada.

(f) Si la memoria de programa de se cambia cuando una secuencia de programa se escribe en el programa de memoria se puede solucionar el modo de arranque y el PLC es encendido o reseteado, el modo de operación boot debe ser seleccionado.
SI el LED de "BOOT" está encendido en el panel frontal del modelo QCPU de altas prestaciones, se ha seleccionado el modo de operación de arranque.

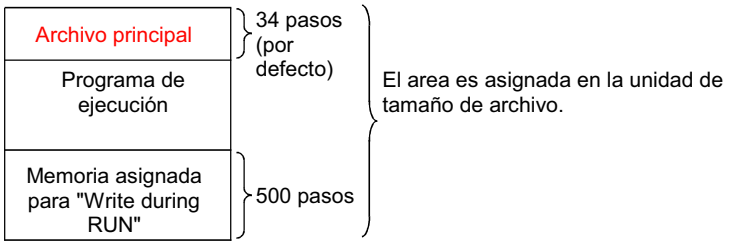
Cancelar el modo de funcionamiento de arranque con el siguiente procedimiento.
1) Escritura de parámetros, en los cuales ningún archivo de configuración de arranque se haya realizado en la memoria de programa.
2) Utilizando el interruptor DIP del módulo CPU configurar “program memory” para la configuración de dispositivo válida. (Configuración de interruptor DIP: SW2: OFF. SW3: OFF)
3) Apagar y encender el PLC o resetear el módulo CPU.
(Después del procedimiento, la configuración dada en los pasos 1) y 2) pasarán a ser válidas.)

6.8 Configuración de archivo de programa

(1) Configuración de archivo de programa

(a) Un archivo de programa consiste en un archivo principal, un programa ejecutable y una asignación de memoria asignada para "Write during RUN".

Configuración de archivo de programa


Archivo principal
Programa de ejecución
Memoria asignada para "Write during RUN"
34 pasos (por defecto)
500 pasos
El area es asignada en la unidad de tamaño de archivo.

(b) Tal y como se muestra seguidamente, el tamaño de un archivo de programa almacenado en el modelo QCPU de altas prestaciones incluye todos los siguientes componentes.

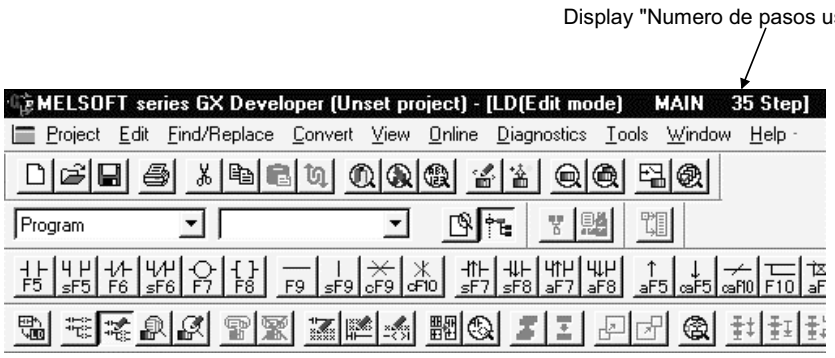
1) Archivo principal: El nombre del archivo, tamaño del archivo y los datos de creación del archivo, etc., son almacenados en esta área.
El tamaño del archivo principal varia entre 25 y 35 pasos (100 y 140 bytes) dependiendo del dispositivo configurado en la ventana de diálogo PLC parameter.

2) Programa de Ejecución: El programa creado es almacenado en esta área
1 paso son 4 bytes.

3) Memoria almacenada para "Write during RUN": Esta área es usada cuando se escribe durante RUN el cual incrementa el número de pasos y es ejecutado desde el software GX Developer.
El valor por defecto es de 500 pasos de programa (2000 bytes).
El número de memoria asignada para "Write during RUN" puede ser cambiado usando la escritura online al program de PLC.
La cantidad de memoria asignada mediante "Write during RUN" puede ser redefinida si la cantidad de memoria asignada no es suficiente para la escritura durante RUN.

(2) El tamaño del programa monitorizado mediante GX Developer

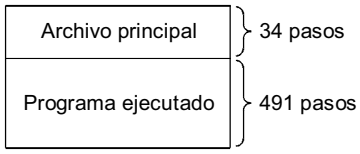
Durante la programación del GX Developer, el tamaño del programa (el total del tamaño de archivo principal y el número de pasos de programas creados) es representado como número de pasos como se muestra seguidamente.
Durante la programación, el tamaño del programa creado es monitorizado.


Display "Numero de pasos usados"
MELSOFT series GX Developer (Unset project) - [LD(Edit mode) MAIN 35 Step]
Project Edit Find/Replace Convert View Online Diagnostics Tools Window Help
Program

PRECAUCIONES

1) La capacidad de programa visualizada durante la programación con GX Developer es la suma de archivos cabecera y capacidad de programa ejecutable y no incluye la capacidad de pasos asegurados para la escritura durante RUN.

(Ejemplo) La capacidad del programa cuya parte del programa de ejecución tiene 491 pasos es visualizada en el GX Developer como se muestra seguidamente. (El archivo principal es fijado a 34 pasos.)



Visulizado en el GX Developer:
34 pasos + 491 pasos = 525 pasos

Estado del archivo en GX Developer

2) Desde que un archivo es almacenado en la memoria de programa en una unidad de tamaño de archivo, la capacidad de programa monitorizado durante la programación con el GX Developer puede diferir con la capacidad del archivo de programa en el modelo QCPU de altas prestaciones.
Referir a la sección 6.9.3 para más detalles.

6.9 Operación de archivos de GX Developer y Precauciones de utilización

6.9.1 Funcionamiento del archivo

El funcionamiento On-line del GX Developer permite a los archivos que son almacenados en memoria, ROM y tarjeta de memoria a efectuar las funciones de la tabla adjunta.
Pero las funciones de archivo disponibles varían de acuerdo con la presencia o ausencia del password (registrado mediante GX Developer), el ajuste de cambio de estado del modelo QCPU de altas prestaciones en "protección contra escritura" y el estado RUN/STOP del modelo QCPU de altas prestaciones.

Tabla 6.5 Funciones de archivos desde GX Developer

Función de archivo	Funcionamiento habilitado/ deshabilitado				Descripción de funcionamiento
	A *	B *	C	D	
Lectura desde PLC	○	△	○	○	Los archivos son leídos desde la memoria.
Escritura a PLC	△	△	×	○	Los archivos son escritos a la memoria de programa y tarjeta SRAM.
Verificación con PLC	△	△	○	○	Verificar la tarjeta de memoria y los archivos de GX Developer.
Escritura de memoria de programa a ROM	○	○	×	○	Escribir una serie de archivos desde el programa de memoria a la ROM estándar o la tarjeta Flash.
Escritura del PLC (Flash ROM)	△	△	×	○	Escribir una serie de archivos desde la memoria de programa a la ROM estándar o la tarjeta Flash.
Eliminación de datos del PLC	△	△	×	×	Se elimina un archivo almacenado en memoria es eliminado.
Formateo de memoria de PLC	○	○	×	×	El formateo de memoria es ejecutado.
Organización de memoria de PLC	○	○	×	×	Los archivos de memoria que no se encuentran contiguos son reorganizados para hacerlos contiguos.
Escritura durante RUN en modo ladder	△	△	×	○	Escribir los cambios realizados en el modo ladder dentro de la memoria de programa.

○ : Ejecución habilitada, △ : Ejecución habilitada con ciertas restricciones, × : Ejecución deshabilitada

Observaciones

1) Los códigos (A, B, C, D) usados en el punto de "función de habilitación / deshabilitación" son explicados en la siguiente tabla.

Tabla 6.6 Función de habilitación / deshabilitación

Código	Descripción
A	Cuando el password de “prohibición de escritura” es registrado en un archivo.
B	Cuando el password de " prohibición de lectura / escritura" es registrado en el archivo
C	Cuando en el modelo de alto nivel QCPU's el interruptor "system protect" se conecta a ON
D	Cuando el modelo QCPU de altas prestaciones está en estado RUN

2)*: La ejecución se permite solamente cuando se introduce el password.

6.9.2 Precauciones de utilización de archivos

(1) Alimentación a OFF (o reset) durante el funcionamiento del programa

(a) Si la alimentación es conmutada a OFF durante el acceso a un archivo que no causará un cambio en los datos, el dato de memoria no se perdará.

(b) Si los archivos y los datos en la memoria del modelo QCPU de altas prestaciones son mantenidos usando la bateria (Q6BAT), el programa de datos de memoria no se perderán cuando la alimentación conmute a OFF durante las siguientes funciones en que causan el cambio de archivos.
- Cambio de tamaño de archivo
- Organización de memoria de PLC
- Nueva creación de archivo
- Escritura de un programa de archivo durante el estado RUN
- Escritura de un progrma en exceso de memoria asignada por la “Escritura durante RUN.
- Leyendo un archivo con la instrucción PLOAD

Los archivos almacenados en la tarjeta de memoria no se perderán a menos que la tarjeta de memoria sea extraida del modelo QCPU de altas prestaciones mientras la alimentación está a OFF.

Comentarios

Si las funciones comentadas anteriormente se cumplen, los datos de medio procesado serán almacenados en la memoria interna del modelo QCPU y será almacenado cuando la alimentación conmute a ON otra vez. Se requiere una batería de seguridad para guardar los datos de memoria interna.

(2) Escritura simultanea al mismo archivo desde múltiples GX Developer’s

El modelo QCPU no permite el acceso desde otros GX Developer’s al archivo que está siendo escrito.

No se permite escribir desde otros “GX Developer’s”. Tampoco está permitido escribir en el archivo que está siendo accedido.

Por lo tanto, para representar la escritura desde múltiples GX Developer para el mismo archivo, empezar el procesado del siguiente GX Developer después de que el procesado del GX Developer actual se complete.

(3) Acceso simultáneo a diferentes archivos desde múltiples GX Developer’s

El modelo QCPU permite accesos simultáneos desde otros GX Developer’s hasta 10 diferentes archivos del mismo módulo CPU.

Comentarios

Para más detalles acerca de la instrucción PLOAD, referirse a la QCPU (modo Q)/Manual de programación QnACPU (Common Instructions).

6.9.3 Capacidad de archivo

El tamaño de archivo varía con el tipo de archivo usado. Cuando se utiliza la memoria de programa, RAM estándar, ROM estándar y tarjeta de memoria, calcular el tamaño del archivo mediante la referencia de la tabla 6.7 mostrada seguidamente.

Tabla 6.7 Lista de Capacidad de archivo

Función	Tamaño de archivo estimado (en bytes)
Drive header	64
Parámetros	Por defecto: 564 (incrementado por el ajuste del parámetro) Para referencias: Cargar la configuración a 70 + (18 × (Número de archivos)) Con la configuración MELSECNET/H como máximo 4096 / unidades incrementadas Con la configuración Ethernet a un máximo de 922 / unidades incrementadas Con configuración CC-LINK a un máximo de 251 / unidades incrementadas Con la configuración de password remoto a 64 + 20 + (número de módulos de tarjeta × 10), incrementada hasta 164
Secuencia de programa	136 *1+ (4 × ((número de pasos) + (número de pasos de seguridad para la escritura durante RUN))
Comentarios de dispositivo	74 + (Tamaño total de comentarios de cada dispositivo) • Tamaño de datos de comentario de un dispositivo = 10 + 10250 × a + 40 × b • a: cociente de (Número de puntos de dispositivo) / 256 • b: resto de (Número de puntos de dispositivo) / 256
Inicialización de dispositivo	66 + 44 × n + 2 × (Número total de puntos de dispositivo especificado en el dispositivo de configuración de valor inicial) • n : número especificado de valores iniciales de dispositivo
Área de configuración de usuario	Cuando la configuración en el tiempo de formateo (0 a 3k)
Configuración de cambio On-line multi-block	Cuando la configuración en el tiempo de formateo (0/1.25k/2.5k)
Registro de archivo	2 × (Número de puntos de archivo de registro)
Dato de señal de muestreo	362 + (20 + 2 × (Número de puntos de palabra de dispositivo) + (Número de puntos de bit de dispositivo) / 8) × (Número de señales) + 12 × (Rango de dispositivo) *2
Dato de historia de errores	72 + 54 × (Número de fallos almacenados)
Dispositivo Local	70 + 6 (Tipo de dispositivo especificado) + 2 × ((número total de puntos de M y de V) / 16 + (puntos D) + 18 × (Puntos totales de T, ST y C) / 16) × (número de programas) *1 • Símbolos "M, V, D, T, ST y C" .se refieren a los siguientes dispositivos: M: relé interno V: relé de flancos D: registro de datos T: temporizador ST: temporizador relativo C: contador

*1: 136 es el valor por defecto (se incrementa dependiendo de los parámetros de configuración).

*2: redondear las porciones de fracción de (dispositivos bit)/8, (número total de puntos M y V)/16 y (número total de T, ST y puntos C)/16.

6.9.4 Capacidad de Memoria para archivos

La capacidad de memoria para archivos indica una capacidad después de que los archivos han sido escritos en le área de memoria.
Para los archivos transferidos desde la tarjeta de memoria al programa de memoria en ambas operaciones, notar que la capacidad de memoria asegurada cambia después de la transferencia.

(1) Unidad de tamaño de archivo.

Cuando los archivos se escriben en el área de memoria, la unidad de capacidad de almacenamiento cambia dependiendo del módulo de CPU y el área de memoria para las cuales los archivos serán escritos. Dicha unidad es la unidad de tamaño de archivo.

(a) Las unidades de tamaño de archivo clasificadas mediante areas de memoria.
Las siguientes tablas indican las unidades de tamaño de archivo clasificadas por el módulo de CPU y el área de memoria para los archivos que serán escritos.

Área de memoria / Tipo de Módulo CPU	Unidades de tamaño de archivo de Memoria de Programa/ Estándar ROM / Memoria Flash ＊1
Q02CPU	128 pasos/512 bytes ＊2
Q02HCPU	
Q06HCPU	
Q12HCPU	256 pasos/1024 bytes ＊2
Q25HCPU	512 pasos/2048 bytes ＊2

＊1: Las unidades de tamaño de archivo de una tarjeta flash se aplican en el caso donde el contenido de la memoria de programa es escrito en la tarjeta Flash mediante el módulo CPU usando el GX Developer.
＊2: 1024 pasos/4096 bytes para el módulo CPU cuyos cinco primeros dígitos del número de serie son "04121" o anteriores.

(b) Unidades de tamaño de archivo clasificadas mediante tarjeta de memoria.

Tipo	Nombre del modelo de la tarjeta de memoria	Capacidad de la memoria
Tarjeta SDRAM	Q2MEM-1MBS	512 bytes
	Q2MEM-2MBS	1024 bytes
Tarjeta Flash	Q2MEM-2MBF	1024 bytes
	Q2MEM-4MBF	1024 bytes
Tarjeta ATA	Q2MEM-8MBA	4096 bytes
	Q2MEM-16MBA	4096 bytes
	Q2MEM-32MBA	2048 bytes

＊1: Las unidades de tamaño de archivo de la tarjeta Flash aplicadas para cualquiera de los siguientes casos.
1) Donde los archivos son escritos en la tarjeta Flash mediante el módulo CPU usando GX Developer.

2) Donde los archivos son escritos directamente en la memoria Flash sin pasar por la CPU.

(2) Ejemplo de cálculo de capacidad de memoria

Cuando los archivos son escritos en el área de memoria, la unidad de capacidad de almacenamiento cambia dependiendo del módulo de CPU y el área de memoria para los cuales los archivos serán escritos. Esta unidad se denomina unidad de tamaño de archivo.

(a) Condiciones

1) Tarjeta de escritura del módulo de CPU: Q25HCPU

2) Archivos escritos

Nombre del archivo	Capacidad de archivo
PARAM. QPA (archivo de parámetro)	564 bytes
MAIN. QPG (secuencia de programa)	525 pasos/2100 bytes ＊1

＊1: Indica la capacidad de programa (archivo principal + programa ejecutado) visulizado en el GX Developer. (Referirse a la sección 6.8)

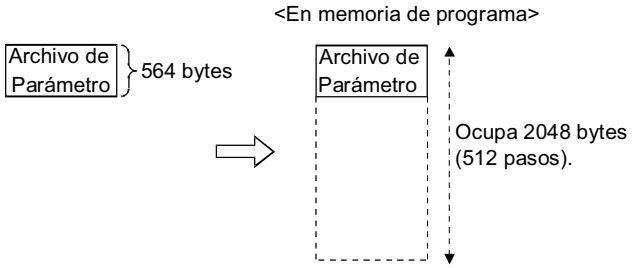
3) Pasos de programa asegurados para la escritura durante RUN: 500 pasos/2000 bytes

(b) Cálculo de capacidad de memoria

El cálculo de la capacidad de memoria se basa en la unidad de tamaño de archivo de la CPU de escritura. En el ejemplo, la unidad de capacidad de archivo de la Q25HCPU es de 512 pasos/ 2048 bytes de acuerdo con la sección (1).

1) Cálculo de parámetro de capacidad de archivo.

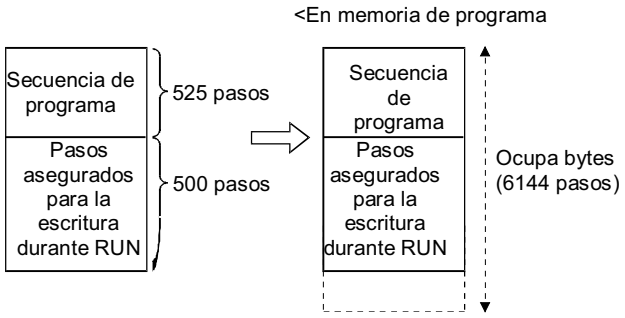
El parámetro de capacidad de archivo es de 564 bytes. Pero, como el parámetro se almacena en la unidad de tamaño de archivo en la memoria de programa, ocupa 512 pasos/2048 bytes.



2) Cálculo de la capacidad de programa

La capacidad de programa es la secuencia de capacidad de programa + pasos asegurados para la escritura durante RUN.

En el ejemplo, la capacidad de programa es 525 pasos + 500 pasos = 1025 pasos. Pero, como el archivo de programa es almacenado en la unidad de tamaño de archivo en la memoria programa, este ocupa 1536 pasos/6144 bytes.


<En memoria de programa
Secuencia de programa
525 pasos
Pasos asegurados para la escritura durante RUN
500 pasos
Secuencia de programa
Pasos asegurados para la escritura durante RUN
Ocupa bytes (6144 pasos)

3) Resultado del cálculo

Nombre del archivo	Capacidad del archivo		Capacidad de memoria
PARAM.QPA	564 bytes		512 pasos (2048 bytes)
MAIN.QPG	Capacidad de secuencia de programa	525 pasos	1536 pasos (6144 bytes)
	Pasos para asegurar la escritura durante RUN	500 pasos	
	Total	1025 pasos	
Capacidad total de memoria			2048 pasos (8192 bytes)

Observaciones

(1) El archivo, el cual fue ejecutado en la CPU cuyo número de serie es "04122" o posterior, podría no ser ejecutado en la CPU cuyo número de serie es "04121" o anterior.

(2) Por lo que respecta a la combinación de escritura a PLC y GX Developer. Las siguientes tablas indican las combinaciones del módulo CPU y versiones de GX Developer usadas cuando los archivos son leídos desde el módulo CPU al GX Developer y escritas hacia el otro módulo CPU.

Leyenda ◎: Escritura a PLC habilitada ○: Restricciones en escritura a PLC

Fuente de escritura CPU / Destino de escritura CPU	GX Developer version8		GX Developer version7	
	Archivos ejecutados en CPU cuyo número de serie es "04122" o posterior	Archivos ejecutados en CPU cuyo número de serie es "04121" o anterior	Archivos ejecutados en CPU cuyo número de serie es "04122" o posterior	Archivos ejecutados en CPU cuyo número de serie es "04121" o anterior
Número de serie "04122" o posterior	◎	◎	○ *2	○ *2
Número de serie "04121" o anterior	○ *1	◎	○ *1 *2	○ *2

*1: Desde que la unidad de tamaño de archivo difiere, los archivos pueden no ser almacenados en la CPU dependiendo de la capacidad de archivo.

*2: A no ser que los pasos asegurados para la escritura durante RUN decrezcan, los archivos pueden no ser almacenados en la CPU dependiendo de la capacidad de archivo.

MITSUBISHI ELECTRIC

Su Socio de Futuro Presente en Todo el Mundo

- Red de ventas
- Fabricas

mitsubishielectric.es

En nuestra página WEB, usted encontrará la última información a cerca de cada uno de nuestros productos y las más recientes novedades. Continuamente actualizamos nuestra página para que usted esté completamente informado.

industrial@sp.mee.com

Puede contactarnos vía email, donde responderemos inmediatamente a sus consultas.

FAX-RESPUESTA 93 589 15 79

Nuestro sistema FAX RESPUESTA le permitirá realizar cualquier consulta y nosotros le enviaremos la información que usted solicite.

LINEA DE ATENCIÓN AL CLIENTE
902 131121

En caso de tener cualquier tipo de problema, puede contactar con nuestra LINEA DE ATENCIÓN AL CLIENTE donde puede contar con ayuda profesional.

Especificaciones sujetas a cambio sin previo aviso
Art.190122 Impreso en España 3/04

MITSUBISHI ELECTRIC EUROPE, B.V. Sucursal en España
División: FACTORY AUTOMATION